UNIVERSIDAD POLITÉCNICA DE MADRID

Escuela Técnica Superior de Ingenieros Industriales

Departamento de Ingeniería Mecánica y Fabricación

Metodología para la Generación Automática de Entornos Virtuales en Simuladores de Conducción Terrestre

TESIS DOCTORAL

Autor

Ing.Carlota Tovar Pérez

Director

Dr. Ing. José María Cabanellas Becerra

MADRID, 2013

Las cosas al final siempre acaban bien
y si no acaban bien,
es que no es el final.

AGRADECIMIENTOS

Nunca me ha gustado este apartado porque considero muy difícil expresar en tan pocas líneas todo el agradecimiento que siento hacia tantas personas. Me encantaría tener espacio para dar gracias por tantos y tantos buenos recuerdos que me llevo de estos años de Tesis, pero como no es posible, intentaré condensarlo lo mejor que pueda.

En primer lugar comenzaré con mi director de Tesis, José María, una persona brillante con un corazón de oro. Gracias por haber estado ahí siempre en todo lo que he necesitado. Gracias por compartir conmigo todas tus ideas geniales que a nadie más se le ocurrirían. A tu lado he aprendido "infinito" y lo mejor de todo es que siempre lo he hecho disfrutando. Cuanto nos hemos reído. Cuantos buenos momentos.

En segundo lugar mis agradecimientos a Carlos Vera, porque sin él no habría comenzado a trabajar en Citef y nada de esto hubiese sido posible. Allá donde estés, gracias Carlos.

En tercer lugar mis agradecimientos a Jesús Félez y José Manuel Mera, porque sin ellos todos los proyectos que me han ayudado a desarrollar esta Tesis no habrían existido. Gracias Jesús por tus numerosos empujes que me han animado a finalizar esta Tesis. Gracias Jose y Luismi por vuestra comprensión en esta recta final de Tesis que ha exigido toda mi dedicación.

Gracias a Martita y Gines por haberme ayudado tanto en el desarrollo de todos estos proyectos. Gines, ha sido una maravilla trabajar contigo.

Gracias a todos los grupos con los que he compartido trabajo en este tiempo, o como diría en la Tesis, gracias a todos los "subsistemas involucrados en la simulación". Gracias a motor gráfico: Joaquín, gracias, gracias y más gracias. Han sido muchos años juntos y muchos buenos momentos. Gracias al sector de modelado: Antonio, gracias a ti y a todo tu equipo por vuestra ayuda. Seguro que el GIS trae una nueva era. Gracias al módulo de conducción: Eduardo, Miguel, Pablo, Ivan, ha sido un placer trabajar con vosotros.

Goyo, que habría sido de mí sin tu ayuda en los congresos, gracias. Mery, muchas gracias por tu paciencia y ayuda con todos los trámites.

Juan David, has hecho que esta última fase haya sido mucho más fácil. Gracias por haber estado pendiente de mí en todo momento.

Carlos, que te voy a decir, sobran las palabras. Sin ti, sin Carlitos y Javi, nada habría sido igual.

Family, friends, Carlos…si no fuese porque me haceis tan feliz nada de esto tendría sentido. Esta Tesis va por vosotros.

ÍNDICE DE CAPÍTULOS.

ÍNDICE DE FIGURAS.

ÍNDICE DE TABLAS.

RESUMEN DE LA TESIS.

Está Tesis refleja los trabajos de investigación que tienen como resultado el desarrollo de una metodología cuyo objetivo es la automatización optimizada en la generación de grandes entornos virtuales presentes en simulaciones de conducción terrestre guiada, es decir, sobre trayectorias predefinidas, destinadas al aprendizaje y entrenamiento de conductores. En ella se aborda el ciclo completo de generación de un entorno virtual, aportando soluciones optimizadas en cada una de las fases.

Para definir estas soluciones, se ha llevado a cabo un estudio en profundidad de las características y requisitos exigidos a este tipo de representaciones virtuales así como de las posibles vías resolutivas, concluyéndose con el establecimiento de tres fases constructivas.

La primera fase, el Modelado Conceptual, realiza una abstracción del mundo real con el fin de automatizar y optimizar su generación. Para ello, tras sintetizar las características que definen a este tipo de entornos así como sus condiciones perceptivas, propone una metodología constructiva que saca el máximo partido de las mismas de una manera hasta ahora no considerada en toda su potencialidad: la creación del entorno mediante el ensamblaje y repetición de un número finito de patrones repetitivos. Son múltiples las ventajas aportadas por este sistema de patrones: escalabilidad, ya que con un número finito de patrones pueden generarse escenarios tan grandes como se deseen; reducción drástica de la memoria de almacenamiento y de los tiempos de carga; disminución de las labores de modelado; facilitación de las tareas de creación y edición de los escenarios.

La segunda fase, el Modelado de Datos, materializa el Modelo Conceptual en un conjunto de estructuras de datos y reglas de generación. Emplea para ello un paradigma orientado a objetos cuyo objetivo es favorecer la reusabilidad de estos objetos tanto dentro de un determinado escenario como entre distintos escenarios virtuales. El elevado volumen y heterogeneidad de la información a manejar hace necesario el desarrollo de métodos que automaticen el procesamiento de la misma. La presente Tesis desarrolla un sistema consistente en varios criterios de organización, corrección y almacenamiento de la información de partida. Dicho sistema permite por un lado la construcción fácilmente escalable y editable de entornos que cumplan las normativas circulatorias vigentes y por otro lado garantiza un óptimo flujo de la información entre los diversos subsistemas integrantes de la simulación. Flujo que ha sido plasmado en la elaboración de diversos protocolos de comunicación.

La tercera fase, el Modelado Físico, desarrolla toda la algorítmica necesaria para concretizar el Modelo de Datos en la representación virtual de un determinado escenario, generando así la Base de Datos del Simulador. Esta Base de Datos suministrará a los diversos subsistemas involucrados en la simulación (Infografía, Motor Gráfico y Módulo de Conducción) toda la información del entorno que requieren para su correcto funcionamiento. Esta algorítmica engloba el posicionamiento 3D de la base de datos visual, su jerarquización y optimización escénica.

La presente Tesis ha desarrollado una serie de algoritmos de posicionamiento modular que garantizan el correcto acoplamiento de los módulos a lo largo de una serie de líneas directrices. Dichas líneas son creadas a partir de la definición de las trayectorias circulatorias, lo que ha permitido a su vez, la definición por parte de esta Tesis de un sistema de niveles de detalle discretos pseudo-variantes con el punto de vista, que permite optimizar la carga geométrica del escenario, mediante la definición en tiempo de precarga de los posibles niveles de detalle de cada módulo en función de su distancia transversal a la trayectoria.

Por otro lado, con el fin de potenciar las ventajas de este sistema de patrones esta Tesis propone el empleo de una serie de shaders que deforman los módulos en tiempo real, lo que permite disminuir el número de módulos necesarios en tiempo de precarga y aumenta la versatilidad constructiva y realismo de los escenarios.

Finalmente, esta Tesis organiza todo el escenario en un grafo de la escena (*scene graph*) que busca minimizar el recorrido del mismo con el fin de maximizar las velocidades de refresco y facilitar las labores de edición de los escenarios. En este punto, la construcción modular del entorno es fundamental para alcanzar dichos objetivos.

Todos los planteamientos teóricos expuestos en esta Tesis se han materializado en las correspondientes aplicaciones informáticas que se han validado como herramientas de desarrollo en la creación de grandes entornos virtuales de simuladores actualmente en funcionamiento, como los de Metro de Madrid de las series 7000, 8000, 3000, 9000 y Citadis y en otros experimentales donde también se han implementado los criterios y resultados perceptivos que optimizan estos simuladores.

SUMMARY OF DOCTORAL THESIS.

This Thesis reflects the research work that has culminated in the development of a methodology whose aim is the optimised automatic generation of large virtual environments intended for guided terrain driving simulations; that is to say, driving over pre-defined paths using a PC. The Thesis approaches the complete cycle of virtual environment generation, providing optimised solutions in each of the phases.

In order to define these solutions an in-depth study was conducted of the characteristics and requirements demanded by this kind of virtual representation as well as of the ways to address these aspects and concluding by setting up three construction stages.

The first stage, Conceptual Modelling, performs an abstraction of the real world aiming to automate and optimize its generation. To this end, after synthesising the characteristics defining these kinds of environments as well as their perceptual conditions, puts forward a constructive methodology that makes the most of these conditions in a way that has never been done hitherto: the creation of the environment by means of assembly and repetition of a finite number of repetitive patterns. The advantages of this pattern system are many: scalability, because with a finite number of patterns can be generated as large scenarios as desired; a drastic reduction in the storage memory required and in loading times; less effort required to do the modelling; greater ease in creating and editing the scenes.

The second stage, Data Modelling, materializes the Conceptual Modelling into a set of data structures and generation rules, using an object oriented paradigm intended to facilitate the reutilization of these objects inside a specific environment as much as between other environments. The high volume and heterogeneity of the information to be handled means that methods need to be developed to automate and process this information. This Thesis develops a system consisting of several criteria for organising, correcting and storing the initial information. On the one hand, this system enables environments to be constructed that meet the traffic regulations in force and which are easy to scale and edit. On the other hand, the system guarantees an optimum information flow between the different subsystems comprising the simulation. A flow that has been embodied in the preparation of diverse communication protocols.

The third stage, Physical Modelling, develops all the algorithms needed to specify the Data Modelling into the representation of a specific scenary, generating the Simulator Database. This Database will supply to the rest of the simulation subsystems (Graphic Designers, Graphics Engine and Driving Module) all the information required to its properly operation. These algorithms include the 3D positioning of the virtual database, its hierarchical structure and scene optimisation.

In this stage, this Thesis has developed a set of modular positioning algorithms to ensure the modules are properly coupled through a set of guidelines. These guidelines are created from the definition of the routes, which, in turn, has enabled this Thesis to define a system of discrete, pseudo-variant detail with the viewpoint. This allows optimising the geometric scene loading by

defining in the pre-load time the possible levels of detail for each module depending on its cross distance to the path.

However, with the purpose of boosting the advantages of this instancing system, this Thesis proposes using a set of shaders that deform the modules in real time. This lets the number of modules required in the pre-load time be reduced while increasing the constructional versatility and realism of the scenes.

Finally, this Thesis organises the whole scene into a scene graph that seeks to minimise the path of the scene in order to maximise the refresh times and make it easier to edit the scenes. At this point, the modular construction of the environment is vital to the achievement of these objectives.

All the theoretical approaches set out in this Thesis have been materialised in the corresponding computer applications which have been validated as development tools in the creation of large virtual environments for simulators that are currently in service, like those of the 7000, 8000, 3000 and 9000 series of Metro de Madrid and Citadis, and in other experimental series where the criteria and perceptual results that optimise these simulators have been implemented.

1.INTRODUCCIÓN.

1.1 JUSTIFICACIÓN.

El empleo de simuladores virtuales de conducción como herramienta de aprendizaje y entrenamiento es, a día de hoy, una técnica ampliamente difundida. La creación de entornos virtuales destinados a este tipo de simulaciones supone un gran reto ya que es necesario el cumplimiento de un alto número de exigencias que en ocasiones resultan contrapuestas: representación de entornos de descomunal tamaño; que reproduzcan con realismo los escenarios que un conductor debe controlar; que permitan elevadas velocidades de refresco; que sean intuitivos en su manejo y edición. La consecución simultánea de todas ellas obliga a la determinación de soluciones de compromiso. La elección de una metodología constructiva adecuada que alcance dicho compromiso es un factor clave a la hora de evaluar el éxito de estos proyectos.

Hasta el momento son numerosos los estudios llevados a cabo para analizar los factores que influyen en la efectividad de una simulación de conducción [1]. Sin embargo, no existe ninguna bibliografía de referencia que sintetice dichos factores y los vincule a la elección de una metodología constructiva eficiente. Es más, ni siquiera se ha establecido una definición que permita esclarecer los puntos que ha de resolver dicho proceso constructivo. Por ello, esta Tesis como punto de partida de sus desarrollos comienza definiendo el término *metodología constructiva de un entorno virtual*, como el conjunto de procedimientos que permiten interpretar, procesar, analizar, modelizar, optimizar, representar virtualmente y garantizar la perdurabilidad en el tiempo, de una determinada información de partida. Esto implica el empleo de toda una amalgama de ciencias: geografía, geometría computacional, informática gráfica y realidad virtual. Cada una de ellas ha sufrido en los últimos años un gran desarrollo tecnológico, ofreciendo multitud de soluciones para su gestión. Sin embargo, el hecho de que dichas herramientas no hayan sido optimizadas para satisfacer los requisitos de estas simulaciones implica una serie de hándicaps : elevados tiempos de formación del personal y gran cantidad de trabajo manual complementario que ralentiza y encarece el proceso, es fuente de numerosos errores y dificulta la incorporación de modificaciones en el entorno. Como consecuencia de estas limitaciones se hace necesario establecer una metodología eficiente que defina las etapas del proceso constructivo y que desarrolle un conjunto de herramientas que, apoyándose en las ya existentes, genere toda la información del entorno requerida por la simulación.

De esta manera, las diversas entidades dedicadas al mundo de la simulación de conducción han creado sus propias soluciones. Para ello han desarrollado plataformas multidisciplinares que abarcan todo el proceso de construcción del entorno, ajustándose a las necesidades particulares de su módulo de conducción y motor gráfico. Tal es el caso de Presagis

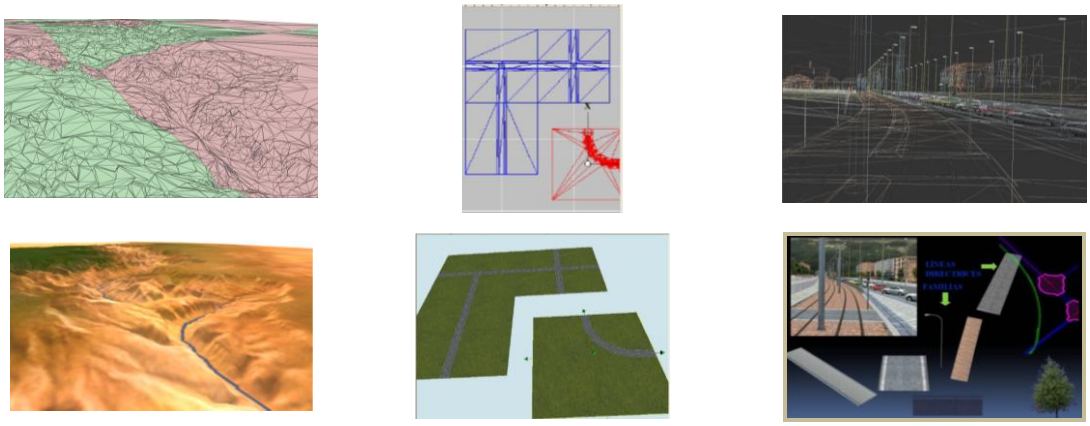
[1] Fisher D.L, Rizzo M, Caird J.K et al. Handbook of Driving Simulation for Engineering, Medicine and Psychology. U.S. CRC Press 2011. ISBN: 978-1-4200-6100-0.

que ha desarrollado para la creación de sus entornos Stage Scenario y de OKTAL, con su herramienta Scaner Studio [2].

Siguiendo esta filosofía constructiva la presente Tesis ha desarrollado la Tecnología Modular, una metodología para la genereación automática de entornos virtuales que ha sido empleada con gran éxito en los simuladores ferroviarios del Centro de Investigación de Tecnologías Ferrroviarias (CITEF) [3]. Esta metodología abarca todo el proceso de construcción de un entorno virtual, combinando de una manera hasta ahora no realizada lo mejor de las dos metodologías constructivas existentes hasta el momento: la construcción mediante mallas ad hoc (Figura 1.1 a) y la construcción mediante losetas repetibles (Figura 1.1 b).

En el primer caso, la prioridad constructiva es la apariencia visual. La gestión de las estructuras de datos que definen el escenario consumen una gran cantidad de recursos, convirtiéndose en una opción inviable en entornos en los que la funcionalidad exigida al mismo sea elevada. Especialmente si dicha funcionalidad proviene del propio entorno, como es el caso de las simulaciones de conducción, caracterizadas por una red de trayectorias circulatorias extensa, que aumenta la complejidad tanto de las estructuras de datos que definen el entorno, como de las funciones que lo manejan.

En el segundo caso, la prioridad constructiva es la funcionalidad. El escenario se genera mediante la concatenación y repetición (término conocido en informática gráfica como instanciación [4], *instancing*) de un número finito de losetas. Gracias a la instanciación, un objeto (llamado objeto Master) que aparezca en diversos lugages del escenario solo necesita cargarse una vez en memoria y podrá reutilizarse tantas veces como sea necesario sin más que emplear una referencia al posicionamiento en memoria del objeto Master. Esto supone un gran ahorro en la memoria de almacenamiento y en los tiempos de carga del escenario. Sin embargo, la rigidez de este sistema da lugar a una escasa flexibilidad constructiva y escenarios poco realistas.



(a) *(b)* *(c)*

Figura 1.1.(a) Entorno generado mediante una malla ad hoc. (b) Entorno generado mediante losetas. (c) Entorno generado por la Tecnología Modular mediante patrones.

La Tecnología Modular aúna lo mejor de ambas metodologías salvando sus inconvenientes. Para ello genera el entorno mediante el ensamblado y repetición de una serie de

[2] Nguyen T, Casas J. "An integrated framework combining a traffic simulator and a driving simulator". Procedia - Social and Behavioral Sciences. 2011. Vol. 20, p. 648-655. ISSN 1877-0428.

[3] http://www.citef.etsii.upm.es/. [Última consulta: 8 Enero 2013].

[4] Shreiner, D. Chapter 3: Drawing with OpenGL. OpenGL Programming Guide: The Official Guide to Learning OpenGL, Versions 4.3, Eighth Edition. Addison-Wesley Professional 2013.ISBN 978-0-321-77303-6.

patrones (Figura 1.1 c). Estos patrones pueden ser de dos tipos: aquéllos que se generan mediante mallas ad hoc y que aunque no se repiten de manera explícita en el escenario obedecen a una serie de reglas de generación que permiten su automatización y los módulos *instanciables*, que representan geometrías que se repiten en el entorno. El grado de instanciación (*instancing*) se puede regular en función de las características de repetibilidad del entorno. De esta manera su creación puede ir desde un entorno plenamente modular a uno generado por completo con mallas ad hoc. Gracias al empleo de este sistema de instanciación la Tecnología Modular logra reducir de manera drástica la memoria de almacenamiento requerida y los tiempos de carga de los escenarios, algo de vital importancia en este tipo de simulaciones. A su vez, la filosofía de patrones empleada en todos sus desarrollos ha proporcionado la flexibilidad y escalabilidad requerida en estos entornos.

1.2 PLANTEAMIENTO.

La Tecnología Modular establece que el éxito de una metodología constructiva para la generación de entornos virtuales destinados a simulaciones de conducción terrestre, dependerá de la correcta resolución de una serie de puntos claves.

El primer punto clave hace referencia al tratamiento de la información de partida y a la gestión de la información generada. A día de hoy, la información disponible para la generación de estas simulaciones se caracteriza por su abundancia, heterogeneidad y ausencia de estándares. (Figura 1.2). Mientras que los altos presupuestos manejados en la simulación militar, han permitido el establecimiento en este área de estándares como CDB (Common Database Specification) y SEDRIS (Source for Environmental Data Representation and Interchange) [5], en el sector civil tal unificación aún no se ha producido. Un intento tuvo lugar con el proyecto europeo TRAIN-ALL, en el cuál diversas entidades dedicadas al mundo de la simulación de conducción se congregaron con el fin de plantear este problema y ofrecer soluciones al mismo. Sin embargo, únicamente se obtuvo como resultado una recomendación sobre el formato a elegir en el diseño de carreteras [6]. Aunque se espera que la Directiva Europea INSPIRE (Directiva 2007/2/CE del Parlamento Europeo y del Consejo, de 14 de marzo de 2007), por la que se establece una infraestructura de información espacial en la Comunidad Europea, resuelva esta problemática en un futuro próximo [7], es una tarea todavía pendiente.

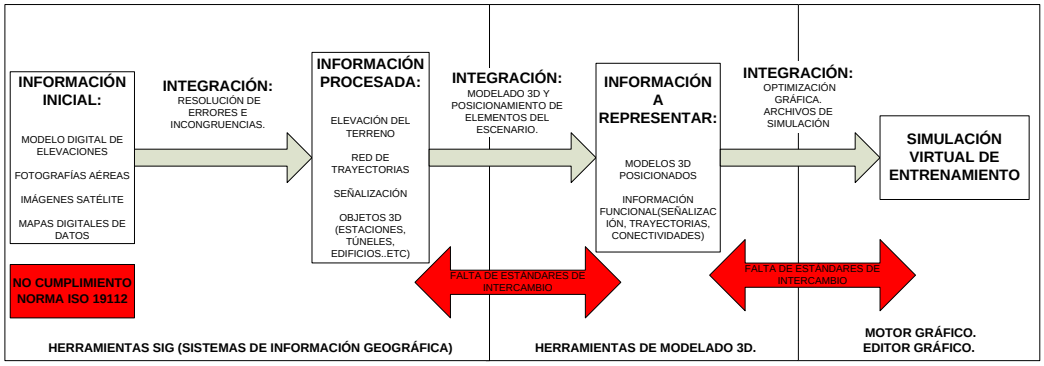


Figura 1.2.Proceso constructivo de un entorno virtual para una simulación de conducción.

[5] www.sedris.org. [Última consulta: 8 Enero 2013].

[6] Chaplier, J., Nguyen That, T., Hewatt, M., Gallée, G. Toward a standard: RoadXML, the road network database format architecture. In proceedings of the Driving Simulation Conference 2010, pp. 211-220.

[7] http://inspire.jrc.ec.europa.eu/documents/Data_Specifications/INSPIRE_DataSpecifi_cation_TN_v3.0.pdf. [Última consulta: 8 Enero 2013].

Por otro lado, son varios los subsistemas involucrados en la simulación que han de hacer uso de esta información: el módulo destinado a la conducción, el motor gráfico y el sector infografista (Figura 1.3). Cada uno de ellos con unas necesidades y objetivos muy distintos.

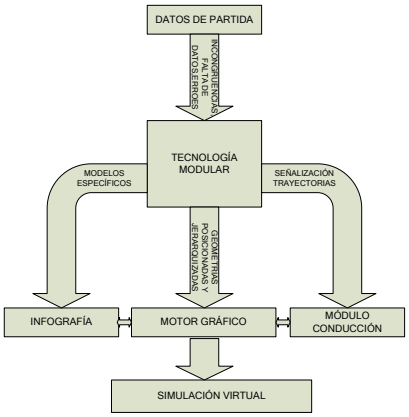


Figura 1.3.Subsistemas involucrados en la generación de un entorno virtual destinado a una simulación de conducción.

La compatibilización de todas sus exigencias garantizando en todo momento la coherencia de la información es una ardua tarea. Los CAD y los Sistemas de Información Geográfica (SIG) se ofrecen como herramienta de ayuda, sin embargo, la especificidad de requisitos que reúnen este tipo de simulaciones, hacen que estas herramientas no sean suficientes por sí mismos. Es necesario el desarrollo de algoritmos y herramientas específicas que ayuden a la automatización de dicha tarea. Esta es la solución adoptada por la Tecnología Modular, que ha desarrollado un Módulo de Generación Topológica, un Módulo de Ajuste Geométrico y una serie de protocolos de comunicación entre los diversos subsistemas integrantes de la simulación que resuelven toda esta problemática

El segundo punto clave hace referencia a la necesidad de representar virtualmente escenarios enormes dotados de una gran funcionalidad garantizando siempre una conducción idónea. Es decir, una elevada velocidad de refresco y la inexistencia de elementos que puedan perjudicar la concentración del conductor, como puede ser la aparición brusca de objetos en el escenario (*popping*). Para ello es necesaria una selección adecuada de las técnicas de optimización gráfica empleadas. En este sentido hay que destacar las grandes capacidades de cálculo de las actuales GPUs (*Graphics Processing Unit*), que han permitido liberar de su carga a la CPU (Figura 1.4). Esto es algo de vital importancia en este tipo de representaciones virtuales, dónde el módulo de conducción consume gran cantidad de recursos. A su vez la posibilidad de emplear programas ejecutables en la GPU, llamados shaders, ha abierto un sinfín de oportunidades para la optimización del entorno gráfico.

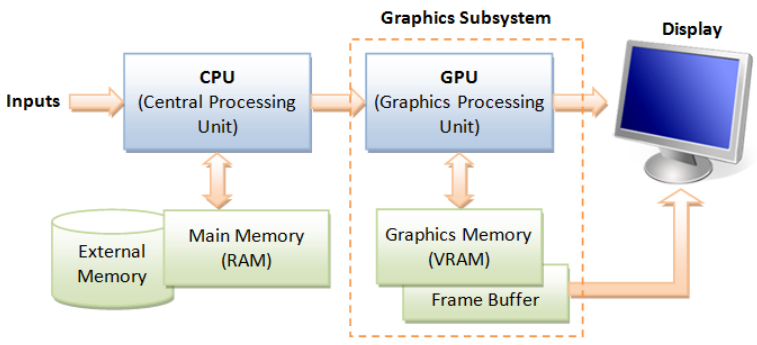


*Figura 1.4.*Arquitectura del hardware gráfico.

Con el fin de sacar el máximo partido posible a las capacidades de las actuales GPUs, los algoritmos de adecuación del nivel de detalle (*level of detail*, LOD), que se encargan de definir en cada instante la representación óptima de cada elemento, han cambiado sus objetivos. Si bien inicialmente, la representación óptima era aquella que minimizaba el número de triángulos que definían la geometría de los elementos a representar, hoy en día las elevadas prestaciones de estas GPUs y el elevado número de triángulos que son capaces de procesar han cambiado esta línea de actuación. Ahora el principal objetivo es minimizar la transacción de información entre CPU-GPU. Esto se ha conseguido agrupando los triángulos en bloques (*batch* o *tile*) [8] , de manera que la optimización se lleva a cabo por bloque (Figura 1.5). Así se consigue disminuir el número de transacciones CPU-GPU (una por bloque en lugar de una por triángulo) a costa de una simplificación geométrica menos óptima, algo que para las actuales GPUs no supone ningún problema.

Figura 1.5.Ejemplo de malla de terreno creada por bloques con distintos niveles de detalle.

Los algoritmos de adecuación del nivel de detalle que calculan en tiempo de ejecución la representación óptima de estos bloques (niveles de detalle continuos, *continuos lod*), emplean un consumo de CPU excesivo para este tipo de representaciones virtuales. Por otro lado, los algoritmos de adecuación del nivel de detalle que calculan la representación óptima de estos bloques en tiempo de precarga (niveles de detalle discretos, *discrete lod*), presentan una serie de inconvenientes:

- Elevado tiempo de procesamiento para generar los bloques, lo que ralentiza una sesión de entrenamiento en la que el conductor conductor necesita crear y visualizar diversos entornos.
- Limitada flexibilidad constructiva, ya que generalmente se trata de losetas rectangulares, que no se adpatan a los trazados circulatorios, lo que dificulta la combinación y reutilización de los mismos.
- Gran consumo de memoria, pues toda la geometría debe ser almacenada en tiempo de precarga.

La Tecnología Modular aporta una solución innovadora que resuelve todos estos problemas apoyándose en una de las herramientas base de la optimización gráfica, la instanciación o empleo de instancias [9]. Para definir dichas instancias, la Tecnología Modular realiza una discretización geométrica y funcional del entorno en un conjunto de familias de módulos. Mediante el el ensamblaje y repetición de dichos módulos a lo largo de una serie de líneas directrices definidas a partir de las trayectorias circulatorias se genera el escenario. Este sistema de instanciación permite disminuir drásticamente los tiempos de carga y la memoria de

[8] Cozzi, P., Ring, K. Chapter 4: Chunked Lod. 3D Engine Design for Virtual Globes. K Peters/CRC Press 2011. ISBN: 978- 1568817118.

[9] Shreiner, D. Chapter 3: Drawing with OpenGL. OpenGL Programming Guide: The Official Guide to Learning OpenGL, Versions 4.3, Eighth Edition. Addison-Wesley Professional 2013.ISBN 978-0-321-77303-6.

almacenamiento necesaria. Por otro lado, mediante al empleo de shaders, la Tecnología Modular deforma estos módulos en tiempo real, disminuyendo así el número de módulos base necesarios para crear el entorno y por tanto la memoria de almacenamiento requerida, dotando al entorno de un gran realismo y versatilidad.

El tercer punto clave hace referencia a la necesidad de facilitar al usuario las labores de entrenamiento: una sesión de entrenamiento implica el lanzamiento de diversos escenarios. Es requisito indispensable por tanto minimizar los tiempos de carga de los mismos. Por otro lado el usuario necesita crear y modificar los escenarios para adaptarlos a los objetivos de cada plan de entrenamiento, lo que exige la creación de herramientas que faciliten dichas tareas. La modularidad ofrecida por esta tecnología, tanto en lo referente al tratamiento y procesamiento de la información de partida, como a la construcción geométrica y jerarquización de la escena, han permitido el desarrollo de un conjunto de herramientas que cumplen todos estos objetivos.

1.3 HIPÓTESIS.

Esta Tesis desarrolla una nueva metodología para la construcción automática de grandes entornos virtuales en simuladores de conducción terrestre guiada bajo PC destinados al aprendizaje y entrenamiento, la denominada Tecnología Modular (Figura 1.6). El desarrollo de esta metodología se sustenta en el establecimiento de las siguientes hipótesis:

- **Hipótesis 1**: Esta Tesis establece la hipótesis de que es posible sintetizar estos entornos en base a un conjunto de patrones repetitivos. Para ello propone y desarrolla criterios y procedimientos que modifican la filosofía constructiva seguida hasta el momento en este tipo de entornos, generando los mismos mediante el ensamblaje y repetición de un conjunto finito de patrones.

- **Hipótesis 2**: La especificidad de requisitos presentes en estos entornos exige el diseño de una metodología que automatice las tareas de introducción, modificación y mantenimiento de la información. Para ello esta Tesis parte de la hipótesis de que es posible sintetizar la construcción topológica de este tipo de entornos en base a la definición de un conjunto de nodos base, a partir de los cuales se genere todo el escenario. Esto exige el desarrollo de un sistema consistente en varios criterios de organización, corrección, procesamiento y almacenamiento de la información de partida.

- **Hipótesis 3**: Por último, esta Tesis establece la hipótesis de que el conocimiento que a priori se tiene de las trayectorias vehiculares, unido a la elección de una construcción modular del entorno, permite la creación de un sistema de niveles de detalles discretos pseudo-variantes con el punto de vista que proporciona las velocidades de refresco necesarias en este tipo de simulaciones. A su vez este sistema resuelve las deficiencias comentadas previamente que suelen acompañar a los sistemas de optimización gráfica basados en el uso de niveles de detalles discretos.

  Para ello esta Tesis propone y desarrolla una metodología que a diferencia de las metodologías convencionales, en las que el terreno es el elemento base y las trayectorias actúan como meras líneas de ruptura, emplea como pilar constructivo la trayectoria. Partiendo del trazado funcional del simulador, la Tecnología Modular crea el resto del entorno mediante el uso intensivo de la instanciación y de la generación automática a partir de perfiles. Este sistema de instanciación se verá completado con el empleo de shaders, lo que potenciará las ventajas de esta metodología. Por otro lado, esta Tesis propone y desarrolla una organización de la geometría y unos protocolos de transmisión de toda la información funcional del entorno que optimiza la rapidez y la coherencia entre los distintos subsistemas del simulador garantizando el poder visualizar la representación virtual en tiempo real y permitiendo la circulación a su través.

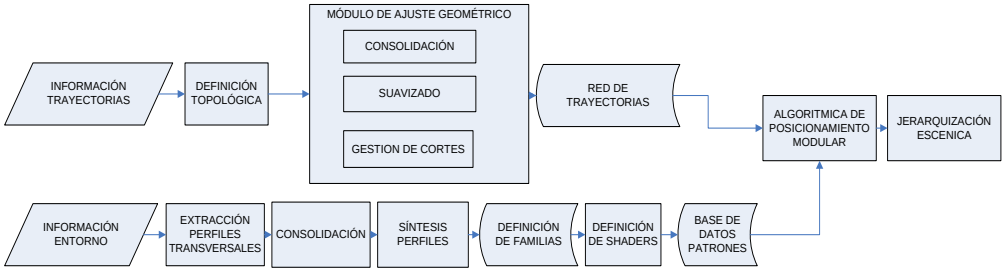


Figura 1.6.Esquema de funcionamiento de la Tecnología Modular.

1.4 OBJETIVOS.

El objetivo de esta Tesis es el desarrollo de una nueva metodología para la construcción automática de grandes entornos virtuales que ofrezca el realismo, la versatilidad, la funcionalidad y las velocidades de refresco que los simuladores de conducción terrestre guiada del CITEF requieren. Esto ha implicado hacer frente a una serie de problemáticas cuya resolución se ha convertido en los objetivos de esta Tesis.

1.4.1 FASE 1: MODELADO CONCEPTUAL.

El Modelado Conceptual, realiza una abstracción del mundo real con el fin de automatizar y optimizar su generación. La dificultad a la hora de generar la base de datos de modelos 3D de un entorno virtual, estriba en alcanzar el grado de optimización exigido por el motor gráfico respetando el grado de calidad visual y funcionalidad exigido a la simulación. Un estudio en percepción visual resulta fundamental antes de decidir qué y cómo se va a modelizar y visualizar. En este sentido, los simuladores de conducción terrestre para el entrenamiento, presentan una serie de características que pueden favorecer en su optimización y a las que no suele sacarse todo el provecho que debiera. El conocimiento a priori de los elementos del entorno que van a jugar un papel fundamental en la representación virtual, así como el de las trayectorias de circulación, las elevadas velocidades de circulación a través del entorno y la repetibilidad de numerosos elementos del mismo, son condiciones que hasta el momento no habían sido explotadas conjuntamente.

El objetivo de la presente Tesis para esta fase ha sido sacar el máximo partido a estas características perceptivas tan particulares de este tipo de entornos, características que por otro lado, suelen pasarse por alto. Para ello explota una herramienta que hasta ahora tan solo había sido empleada con fines constructivos de una manera muy restringida: la instanciación. Previa búsqueda de unos patrones repetitivos del entorno, tanto geométricos como funcionales, la Tecnología Modular discretiza el escenario en un número finito de patrones, que tras ser posicionados y sometidos a una serie de transformaciones se ensamblan meticulosamente.

1.4.2 FASE 2: MODELADO DE DATOS.

El Modelado de Datos, materializa el Modelo Conceptual en un conjunto de estructuras de datos y reglas de generación. La generación totalmente automática de la información inicial requerida para la creación de un entorno virtual destinado a una simulación de conducción terrestre es un objetivo muy ambicioso. Las aplicaciones CAD y SIG se ofrecen como

herramienta de ayuda, sin embargo son múltiples los problemas a los que hay que hacer frente en una representación virtual de estas características. La problemática asociada a la información de partida ha sido sintetizada por esta Tesis en los siguientes puntos:

- Fuentes de información incompletas, incoherentes, erróneas.
- Continuas reestructuraciones y ampliaciones que exigen una vérsatil edición y escalabilidad de los entornos garantizando siempre la integridad de todos los datos.
- El cumplimiento de numerosas exigencias, que a veces entran en conflicto:
  - Exigencias al trazado por parte del subsistema de conducción.
  - Exigencias al trazado por parte de la normativa circulatoria (radios de curvatura y gradientes máximos y mínimos).
  - Exigencias topológicas para garantizar la correcta circulación.
  - Máximas deformaciones longitudinales admisibles.
  - Exigencias por parte del subsistema de motor gráfico: óptima carga gráfica, aspecto visual continuo.

La resolución de estos problemas garantizando el cumplimiento de las restricciones vistas, lleva al desarrollo de una algorítmica muy específica, lo cuál está fomentando la creación de aplicaciones particularizadas creadas a partir de librerías de geometría computacional orientadas a los SIG.

Para solventar los problemas de esta fase, esta Tesis ha clasificado la información necesaria para la construcción del entorno virtual en tres categorías y ha planteado los siguientes objetivos para cada una de ellas:

- **Información topológica**: descripción de las relaciones de interdependencia entre los diferentes elementos existentes en el entorno, siendo principalmente importante la definición de las conectividades entre trayectorias, responsables del correcto movimiento a través del mismo. Para llevar a cabo la definición topológica del entorno, la presente Tesis se ha propuesto como objetivo la creación de un Módulo de Generación Topológica que por un lado cumpla todas las restricciones funcionales y geométricas exigidas por el subsistema de conducción y por otro lado sea de gran versatilidad, de manera que la variedad de posibles entornos circulatorios a construir sea elevada y en el caso de aparecer nuevas funcionalidades, éstas puedan añadirse cómodamente. Para ello esta Tesis ha definido como elemento base el nodo. La diversidad de nodos da lugar a diversos tipos de conexiones, lo que implica definiciones geométricas y funcionalidades específicas. Estos nodos son definidos en base a la normativa circulatoria ferroviaria y vehicular. Por otro lado, este tipo de entornos se ven sometidos a continuas reestructuraciones que se deben incorporar de manera eficiente, sin motivar inconsistencias con las zonas que no es necesario modificar. Esto solo es posible a través de la manipulación automática de una base de datos que guarde todas las relaciones topológicas existentes entre los diversos elementos del entorno.

- **Información geométrica**: incluye la descripción del terreno, con su elevación y texturado, la definición geométrica de elementos lineales (carreteras, ríos, caminos, etc..) y de elementos anejos (árboles, edificios…). Para procesar dicha información la presente Tesis se ha puesto como objetivo el desarrollo de un Módulo de Ajuste Geométrico. Dicho módulo procesa la información de partida mediante algoritmos de consolidación, gestión de cortes y suavizado, obteniendo así la definición del entorno en base a una serie de estructuras geométricas base (biarcos, NURBS, nubes de puntos) que respetan las diversas restricciones a las que se ven sometidos este tipo de entornos: restricciones de la normativa circulatoria, de los diversos subsistemas involucrados (conducción y motor gráfico) y del proyecto específico que se esté desarrollando.

- **Información funcional**: descripción de las características funcionales de las diversas trayectorias (tipos de carriles, de vías…), elementos de señalización (marcas viales, semáforos ..), conexiones (rotondas, cruces..)..etc. El objetivo planteado en este caso por esta Tesis ha sido la síntesis del entorno en un conjunto de familias, cada una de ellas con una funcionalidad específica definida teniendo en cuenta las normativas circulatorias. Esta información se procesa para satisfacer las necesidades del resto de subsistemas involucrados en la simulación: se definen los módulos que componen la familia para su correcto modelado; se estructura en un grafo de la escena (*scene graph*) que permitirá al motor gráfico simular su comportamiento; se posicionará en el entorno de modo que el subsistema de simulación pueda regular correctamente el tráfico.

Finalmente, es necesario definir protocolos que gobiernen la transacción de información entre todos los subsistemas que intervienen en la simulación (motor gráfico, módulo de conducción y sector infografista) garantizando su integridad ante posibles cambios. Las herramientas existentes para este fin son las creadas por las propias entidades dedicadas a la generación de simuladores de conducción, por lo que a la hora de abordar un proyecto o bien se subcontrata a estas entidades o se elaboran herramientas propias. De esta manera, el siguiente objetivo de esta Tesis para esta fase, es el desarrollado de facto para CITEF, de todos los protocolos de comunicación entre los subsistemas que participan en sus simuladores.

1.4.3 FASE 3: MODELADO FÍSICO.

El Modelado Físico, desarrolla toda la algorítmica necesaria para concretizar el Modelo de Datos en la representación virtual de un determinado escenario, generando así la Base de Datos del Simulador. Esta Base de Datos suministrará a los diversos subsistemas involucrados en la simulación (Infografía, Motor Gráfico y Módulo de Conducción) toda la información del entorno que requieren para su correcto funcionamiento. Esta algorítmica engloba el posicionamiento 3D de la base de datos visual, su jerarquización y optimización escénica.

Los diferentes algoritmos de adecuación del nivel de detalle empleados en la representación en tiempo real de grandes entornos virtuales realizan discretizaciones del entorno, empleando para ello distintas primitivas de trabajo. En función de la primitiva elegida, se define una algorítmica de posicionamiento cuyo objetivo será conseguir el acoplamiento de todas las primitivas garantizando un aspecto final continuo tanto temporalmente, es decir, que no existan transiciones bruscas de nivel de detalle , como espacialmente.

En esta Tesis, la primitiva de trabajo es el patrón o módulo. Las trayectorias circulatorias, elemento clave en toda simulación virtual de conducción terrestre, juegan en la Tecnología Modular un doble papel. Por un lado, como en toda simulación de este tipo, guían el tráfico y por otro lado sirven de líneas directrices para el posicionamiento modular.

Puesto que las líneas directrices pueden presentar cualquier definición geométrica, uno de los objetivos que esta Tesis se propone para esta fase es el desarrollo de una Algorítmica de Posicionamiento que mediante el ensamblado y repetición de un número finito de módulos pueda reproducir el entorno garantizando siempre la continuidad espacial del mismo. La continuidad temporal quedará garantizada mediante una adecuada jerarquización en el grafo de la escena. Todos los cálculos necesarios para llevar a cabo el posicionamiento y jerarquización no suponen ningún inconveniente en el consumo de recursos ya que se realizan en la fase de precarga (Figura 1.7).

El segundo objetivo de esta Tesis para esta fase será por un lado disminuir el tamaño de la base de datos visual y como consecuencia los tiempos de carga del escenario y por otro lado, aumentar la versatilidad constructiva. Para ello la Tecnología Modular hace uso de los shaders. Gracias al desarrollo de una serie de shaders el conjunto base de módulos queda reducido a un

único elemento recto, que será curvado y sometido a diferentes tipos de transformaciones garantizando el correcto acoplamiento entre los mismos y el realismo del escenario.

Por último, esta Tesis se propone como objetivo la organización jerárquica de todo el entorno en un grafo de la escena que facilite el recorrido del mismo con el fin de garantizar las velocidades de refresco requeridas y las labores de edición de los escenarios.

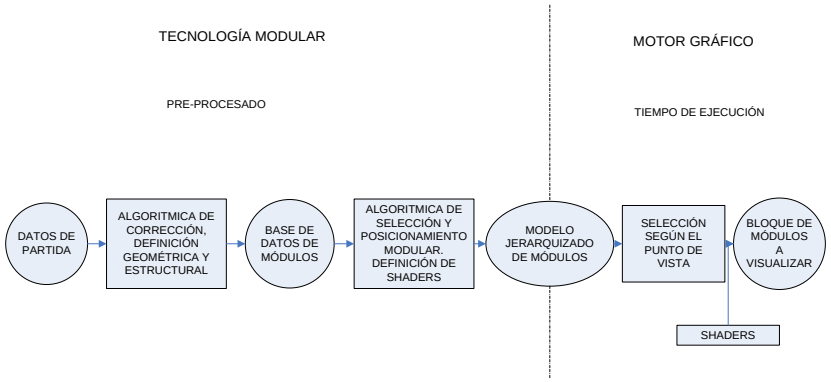


Figura 1.7.Esquema de generación de un entorno virtual seguido por la Tecnología Modular.

1.5 RESUMEN DE CAPÍTULOS.

En el Capítulo 2 se realizará un estudio del estado del arte sobre la generación de entornos virtuales en simuladores de conducción terrestre. Este capítulo dejará patente la necesidad de crear una metodología que abarque todo el proceso de generación de dichos entornos, automatizando al máximo todas las tareas implicadas. Esta necesidad será plasmada por la presente Tesis en la creación de la Tecnología Modular. Esta tecnología, desarrolla una metodología constructiva de entornos virtuales que sintentiza en tres fases: Modelado Conceptual, Modelado de Datos y Modelado Físico que aborda en los restantes capítulos.

El Capítulo 3 tratará el Modelado Conceptual. Se explicarán las ventajas aportadas por la instanciación en toda representación virtual, analizando los requisitos exigidos a un entorno para la aplicación de la Tecnología Modular (Figura 1.8). Las trayectorias actuarán como bloque constructivo básico de la representación virtual, sirviendo de línea base para el ensamblaje y/o como contorno de los entornos circundantes. El carácter predominantemente unidimensional que éstas confieren al entorno, unido a la búsqueda de patrones repetitivos permitirá explotar la instanciación como técnica de optimización base para la generación del entorno de una manera hasta ahora no realizada. Se explicará detalladamente el concepto de módulo y familia de módulos y mallas ad hoc.

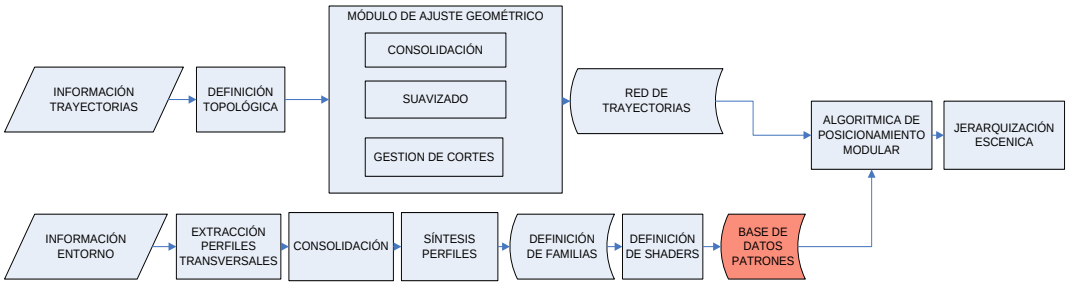


Figura 1.8.Desarrollos de la Tecnología Modular tratados en el Capítulo 3.

El Capítulo 4 se centrará en el Modelado de Datos (Figura 1.9). En primer lugar se tratará el Módulo de Generación topológica y a continuación el Módulo de Ajuste Geométrico desarrollados por esta Tesis para resolver todos los errores e incongruencias que suelen ir asociados a las fuentes de información disponibles para este tipo de proyectos. Se finalizará con la definición de los perfiles transversales que servirán para construir el entorno circundante.

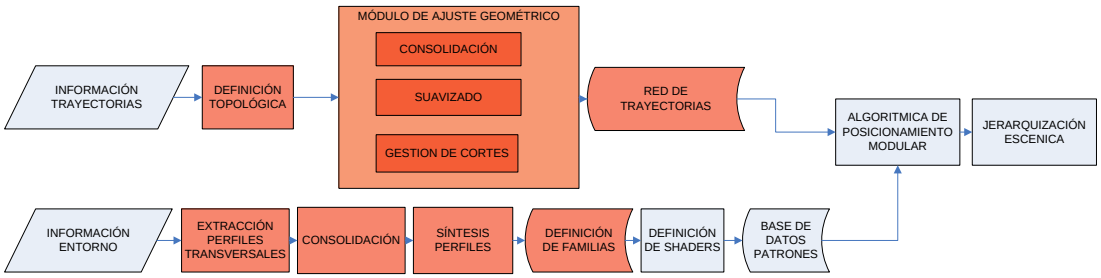


Figura 1.9.Desarrollos de la Tecnología Modular tratados en el Capítulo 4.

El Capítulo 5 tratará el Modelado Físico. En primer lugar se explicará la Algorítmica de Posicionamiento Modular (Figura 1.10). Se estudiarán el número, tipo y posible colocación de módulos para las diferentes definiciones geométricas que el Módulo de Ajuste Geométrico puede proporcionar. A continuación se explicará el concepto de shader y las aportaciones que la Tecnología Modular ofrece gracias a ellos. Finalmente se definirá la estructura del grafo de la escena diseñada en esta Tesis y se explicarán todos los parámetros optimizables por la Tecnología Modular al crear un entorno virtual. Por último se presentarán ejemplos de proyectos en los que la Tecnología Modular ha participado con gran éxito.

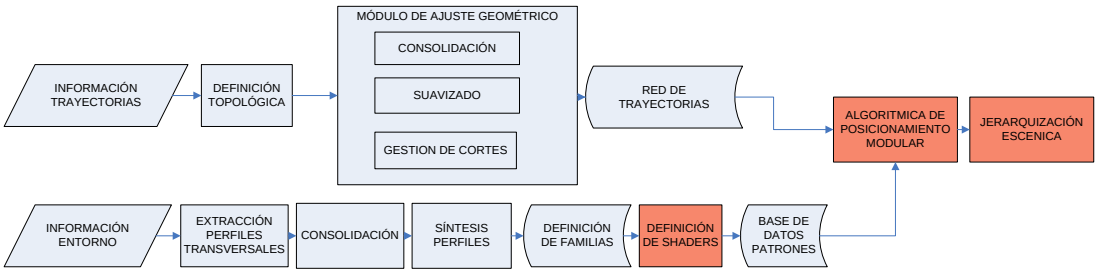


Figura 1.10.Desarrollos de la Tecnología Modular tratados en el Capítulo 5.

El capítulo 6 expondrá las conclusiones y futuras líneas de investigación.

2. ESTADO DEL ARTE.

2.1 INTRODUCCIÓN.

El primer problema que surgió a la hora de afrontar esta Tesis, y que se ha mantenido a lo largo del desarrollo de la misma, ha sido la complejidad de llevar a cabo el estudio del estado del arte. Cuando se comenzó esta investigación la creación de entornos virtuales para simuladores de conducción terrestre consistía en un conjunto de tareas muy diversas e inconexas cuya tecnología estaba en un estado de desarrollo muy temprano. Dichos simuladores estaban iniciando su expansión y no se había dado toda la relevancia que se debiera a la construcción de sus entornos y a las características que los hacían singulares y los diferenciaban de otras representaciones virtuales. Para su construcción se utilizaban un sinfín de herramientas que no guardaban ninguna conexión y a las que les faltaba mucho por madurar. Es así como surge la necesidad de creación de una metodología que, tras estudiar las características y requisitos que se exigen a estos simuladores, estableciera un flujo de actividades a realizar, centralizando y optimizando todo el proceso constructivo de estos entornos. Es así como surge la necesidad de desarrollar esta Tesis.

El estudio del arte que se ha llevado a cabo sobre la manera en que las principales entidades dedicadas al mundo de la simulación de conducción terrestre han generado sus entornos ha sido laborioso. En los inicios de esta Tesis la novedad del tema y los beneficios económicos que reportaban la construcción de estos simuladores hacía que estas entidades se mostrasen recelosas de compartir sus técnicas. Hoy en día, publicadas en la red las diferentes herramientas que han desarrollado con el fin de comercializarlas, se ha podido comprobar cómo esta misma línea de centralización y optimización de tareas por la que optó en sus inicios esta Tesis, ha sido la opción seguida por las principales entidades dedicadas al desarrollo de estas simulaciones, cada una por supuesto con sus particularidades. Esta misma tendencia es la que ha impulsado a diversas herramientas comerciales que intervenían en la creación de estos entornos ha fusionarse para conseguir productos integrados, con formatos compatibles que facilitasen el intercambio de la información entre los mismos.

A su vez, la falta en este sector de unos estándares que regulen el interambio de información, ha llevado al desarrollo de herramientas particulares, que se han ido beneficiando de los diversos avances tecnológicos que se han sucedido en estos años. El aprovechamiento de estos últimos avances ha sido precisamente uno de los objetivos y grandes hándicaps que se ha tenido que afrontar en el desarrollo de esta Tesis: crear una metodología en todo momento puntera y a su vez capaz de perdurar en el tiempo. Una metodología lo suficientemente versátil como para aprovechar los continuos avances del hardware manteniendo la compatibilidad con escenarios generados con anterioridad.

A continuación se expone el estado del arte según se ha llevado a cabo su estudio para la elaboración de esta Tesis. En primer lugar, en este capítulo se realiza una introducción al tema, definiendo las áreas implicadas y haciendo un breve repaso histórico que permita vislumbrar los vertiginosos cambios tecnológicos vividos. Posteriormente se continúa con un estudio del proceso de creación de estos entornos como un todo, analizándose los factores influyentes en la elección de una metodología u otra. Este estudió sirvió para definir una metodología de construcción hasta entonces inexistente en este ámbito.

Finalmente se exponen las soluciones a las que actualmente han llegado las distintas entidades dedicadas al mundo de la simulación de conducción terrestre y se concluye con los simuladores del CITEF, que han puesto en práctica las soluciones desarrolladas por esta Tesis.

En los capítulos posteriores se detallará el estudio del arte realizado para cada una de las fases constructivas propuestas por esta metodología junto a la solución desarrollada por esta Tesis para cada una de ellas.

2.2 LA GENERACIÓN DE ENTORNOS VIRTUALES.

El abaratamiento de costes que está teniendo lugar como consecuencia de los rápidos avances del hardware, ha permitido el empleo de grandes entornos virtuales como herramienta de trabajo en multitud de disciplinas. Simulación, geomorfología, hidrología, ingeniería civil, de telecomunicaciones, planificación urbanística, estudio de impacto ambiental y turismo son ejemplos de sus múltiples aplicaciones [10] [11].

Sin embargo, la generación de grandes entornos virtuales exige la aplicación de toda una amalgama de ciencias, lo que dificulta enormemente la automatización total de este proceso. Fotogrametría, teledetección, cartografía, geometría computacional, bases de datos, informática gráfica…, cada una de ellas ofrece un amplio abanico de recursos a la hora de procesar la información relacionada con el entorno.

En el proceso de gestación de estos grandes escenarios virtuales podemos sintetizar la convergencia de tres áreas bien diferenciadas: sistemas de información geográfica, diseño asistido por computador y realidad virtual. La sinergia conseguida entre estos tres ámbitos ha abierto las puertas hacia la automatización de dicho proceso de generación.

Los sistemas de información geográfica han aportado herramientas para el procesado de la información de partida. Si bien en sus inicios la disponibilidad de equipos informáticos que pudieran hacer frente a los complejos procesos de cálculo involucrados en el análisis de datos territoriales constituía su mayor limitación, hoy en día la disponibilidad de información geográfica ha pasado a ser el gran problema. La adquisición de datos territoriales fiables se ha convertido en la mayor inversión tanto económica como de tiempo.

El diseño asistido por computador [12] ha sido el vínculo que ha permitido la evolución desde las primeras representaciones del terreno a través de mapas digitalizados hasta las actuales representaciones virtuales del mismo. En 1963, Ivan Sutherland, a través de su disertación sobre el programa Sketchpad [13], causó una revolución sin precedentes en el mundo de la computación. Sketchpad supuso no sólo el desarrollo de una de las primeras interfaces gráficas, sino la introducción de multitud de conceptos que sentaron las bases de la informática gráfica. A partir de ese momento la evolución desde lo que inicialmente se llamó cartografía automatizada hasta lo que hoy en día se conoce como entornos virtuales no ha tenido tregua. El diseño asistido por computador permitió las primeras representaciones computerizadas de mapas en dos dimensiones,

[10] Fisher, D.L., Caird, J.K., Rizzo, M. Chapter 1: Handbook of Driving Simulation for Engineering, Medicine, and Psychology: An Overview. Handbook of Driving Simulation for Engineering, Medicine and Psychology. CRC Press 2011. ISBN: 978-1420061000.

[11] Craig, A.B., Sherman W.R. , Will J.D. Chapter 2: Applying Virtual Reality. Developing Virtual Reality Applications: Foundations of Effective Design. Morgan Kaufmann 2009. ISBN: 978-0123749437.

[12] Foley, J.D., VanDamm, A., Feiner, S.K. Chapter 1.Computer Graphics - Principles and Practice. Addison Wesley, second edition.1995. ISBN: 978-0201848403.

[13] Sutherland, I. Sketchpad: A Man Machine Graphical Communication System. PhD thesis at MIT, 1963.

que más tarde se convirtieron en representaciones alámbricas tridimensionales del terreno y finalmente en sólidas superficies 3d.

Las mayores exigencias computacionales implícitas en la representación virtual de estos grandes entornos, impulsó nuevos desarrollos tanto de hardware como de software. La interactividad sobreentendida de estos sistemas, exigía unas elevadas velocidades de redibujado que sólo pudo abordarse con la llegada de un hardware más potente y especializado y con el desarrollo de herramientas software para la optimización de los recursos necesarios.

En el siguiente apartado, un estudio a lo largo de la corta pero intensa historia de este área de investigación pondrá de manifiesto los diversos frentes en los que de una manera simultánea surgió la necesidad de representar grandes entornos virtuales. Unos priorizando un análisis pormenorizado del terreno buscaron una representación 3d exhaustiva de éste, dejando en un segundo plano su manipulación en tiempo real. Otros antepusieron una simulación en tiempo real y se conformaron con un aspecto visual aparente, recurriendo para ello a grabaciones de video.

Los grandes avances tanto del hardware como del software permitieron la convergencia de ambas necesidades. En la actualidad existen numerosas entidades académicas, gubernamentales y comerciales involucradas en el mundo de la simulación virtual, que han ido aportando a lo largo de estos últimos años numerosas herramientas que solventan, de una manera más o menos automatizada, las distintas etapas involucradas en el proceso de gestación de un entorno virtual.

Sin embargo todavía son muchos los frentes de investigación abiertos. La diversidad de fuentes y formatos dificulta las labores de creación de herramientas estandarizadas que automaticen su proceso de consolidación así como el proceso de construcción geométrica 3d del escenario. Las diversas finalidades a las que van destinadas dichas representaciones virtuales, hacen necesario el desarrollo de herramientas particularizadas que permitan la generación de sistemas de información geográfica a medida y algoritmos de optimización de recursos gráficos específicos para cada aplicación.

2.3 HISTORIA

La representación de grandes entornos fue uno de los muchos campos que sufrió un cambio drástico en la década de los 50 con la aparición de los primeros ordenadores programables. Hasta esa fecha se pueden destacar dos ámbitos en los cuales existió la necesidad de representar vastas extensiones de terreno: la cartografía y la simulación aérea. La cartografía disponía para ello de mapas impresos. La simulación aérea, limitada por los escasos recursos tecnológicos, se limitó en sus inicios, alrededor de 1930, a generar simuladores sin retroalimentación visual, cuyo único objetivo era familiarizar al piloto con el panel de mandos y proporcionar una rudimentaria retroalimentación de movimiento. A lo largo de la década de los 30, la compañía Link [14], pionera en este campo, fue aportando novedades. Primero con el cyclorama introdujo la retroalimentación visual: la escena era pintada sobre las paredes de la habitación donde tenía lugar el entrenamiento (Figura 2.1).

[14] Page, R.L. Brief History of Flight Simulation. In SimTechT 2000 Proceedings. Sydney: The SimtechT 2000 Organizing and Technical Committee, 2000.

Figura 2.1.El Link Trainer.

En 1939 Link desarrolla un simulador que permitía cruzar el Atlántico y servir de entrenamiento nocturno (Celestial Navigation Trainer). Sobre la cabeza del piloto se colocaba una cúpula a la que iba fijado un arreglo de luces imitando las constelaciones (Figura 2.2). El movimiento de la cúpula simulaba el cambio de posición de estas constelaciones en función del tiempo y del movimiento del avión, de manera que el piloto pudiese en todo momento determinar su posición a partir de éstas. Para simular el terreno, debajo de la cabina del piloto (Figura 2.3) se colocaban grandes fotografías aéreas sobre plataformas móviles, aumentando así la sensación de realismo y permitiendo practicar tácticas de bombardeo.

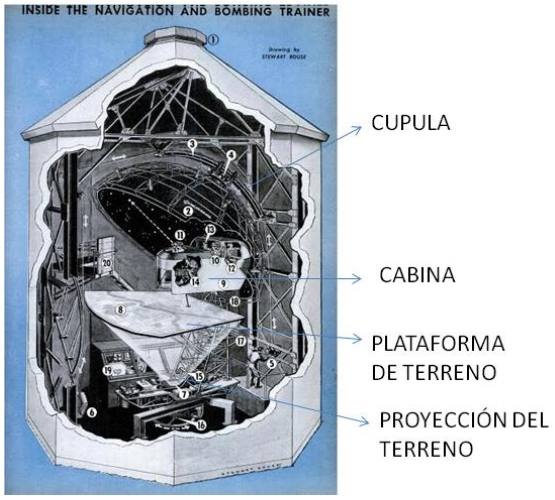


Figura 2.2.El Link Celestial Navigation System, mostrando la cúpula de proyección.

Figura 2.3.Cabina del piloto del Link Celestial Navigation System.

En la década de los 50 los cambios se suceden vertiginosamente. La llegada de los computadores digitales dio un vuelco a la concepción del tratamiento de la información geográfica abriendo dos frentes de investigación distintos. Por un lado los cartógrafos vieron la posibilidad de cambiar su medio de trabajo pasando del formato analógico al formato digital de mapas, dando pie así a la llamada cartografía automática o digital. Por otro lado los servicios de defensa militar, la NASA, organismos de administración gubernamentales, investigadores de las ciencias de la tierra y diversos sectores de negocios vieron en los ordenadores un medio para sustituir su actual herramienta de trabajo: los mapas impresos. Comienza a surgir así la necesidad de los Sistemas de Información Geográfica.

Por otro lado en la década de los 50, se desarrolla el primer ordenador capaz de generar en tiempo real texto y gráficos en la pantalla de un tubo de rayos catódicos: el Whirlwind (Figura 2.4). El Whirlwind se desarrolló con el fin de constituir el primer sistema programable de un simulador de vuelo. El Whirlwind recibía datos de posicionamiento de un avión que le enviaba

una estación de radar situada en Massachussets. La costa este de Massachussets aparecía representada en la pantalla a través de unas serie de puntos y cuando se recibía información sobre el posicionamiento, un símbolo representando el avión aparecía en la pantalla sobre el plano de Massachussets. Robert Everett diseñó un dispositivo de entrada llamado lápiz óptico a través del cual los operadores podían solicitar información sobre el avión. Cuando el lápiz óptico se colocaba sobre la posición del avión, un evento era enviado al Whirlwind que respondía desplegando un texto en la pantalla con la identificación del avión, velocidad y dirección. El Whirlwind fue empleado por los sistemas de defensa de las fuerzas aéreas estadounidenses para su proyecto SAGE (Semi-Automatic Ground Environment) [15].

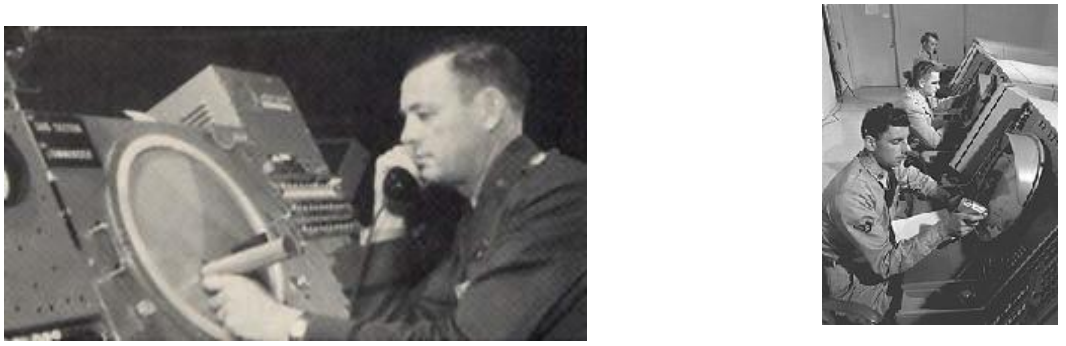


Figura 2.4.Empleo del lápiz óptico (light pen o light gun) en el Whirlwind.

En 1956, Morton Heiling genera la primera experiencia virtual multisensorial, el Sensorama (Figura 2.5). El Sensorama combinaba un película proyectada, sonido, vibración, viento y olores, todo ello con el fin hacer sentir al usuario inmerso en la película, no como un mero espectador de ella. Toda la experiencia era grabada previamente y después reproducida para el usuario. Todavía se estaba muy lejos de una generación automática en tiempo real desde un ordenador.

Figura 2.5.El Sensorama.

Los 60 marcaron un antes y un después en el desarrollo de grandes entornos virtuales, ya que en ésta década tiene lugar el nacimiento de sus pilares: Sistemas de Información Geográfica (SIG), Diseño asistido por computador y Realidad Virtual.

La digitalización de mapas se completó en los 60 con el desarrollo de técnicas de análisis espacial, métodos matemáticos y estadísticos, herramientas para el análisis de redes y una gran

[15] Funding a Revolution: Government Support for Computing Research. Computer Science and Telecommunications Board (CSTB), National Research Council. Washington, DC. National Academy Press,1999. ISBN: 978-0309062787.

variedad de técnicas para el modelado geográfico. Se produce así la completa fusión entre los mapas y los ordenadores: surge el concepto de Sistema de Información Geográfica o SIG. El SIG se definió así como una tecnología capaz de combinar, almacenar, manipular y representar información geográficamente referenciada por ordenador [16]. Pronto surgen las primeras comparativas sobre los diferentes métodos de almacenamiento de la información del terreno [17] . Con la llegada de los mapas digitales, los contornos dejan de ser un formato óptimo para el almacenamiento topográfico, convirtiéndose las mallas regulares en el formato convencional. En esta década la representación de superficies a través de triángulos se realizó sin conseguir una conectividad topológica. Habría que esperar todavía una década más para la aparición de las primeras mallas de triángulos irregulares (TIN, Triangulated Irregular Networks)[18].

En 1964 Howard Fisher funda el laboratorio de informática gráfica y análisis espacial de Harvard (The Harvard Lab for Computer Graphics and Spatial Análisis). El laboratorio se convirtió en uno de los más influyentes centros de investigación siendo pioneros en un gran número de herramientas software para el análisis espacial de datos. En 1966 Howard Fisher desarrolla SYMAP (Synagraphic Mapping System), la primera aplicación para el tratamiento computerizado de mapas [19].

En 1969 Jack & Laura Dangermond fundan ESRI, Environmental Systems Research Institute y Jim Meadlock funda Intergraph Corporation. En este mismo año Ian McHargs publica su libro Design With Nature, siendo pionero en las técnicas de desarrolllo para la superposición de mapas.

Los 60 también fueron decisivos para el Diseño Asisitido por Computador. En 1963, Ivan Sutherland presentó su disertación, " Sketchpad : A Man-machine Graphical Communications System" 20 . Sketchpad supuso no sólo la presentación de una de las primeras interfaces gráficas de usuario sino la introducción de un gran número de conceptos básicos de la informática gráfica (Figura 2.6). Entre sus ideas destacan:

- la necesidad de una estructura jerárquica interna a la hora de representar un objeto por ordenador y la definición de ese objeto en términos de subobjetos;
- el concepto de objeto master y objetos instanciados, que no son más que versiones del objeto master a las que se han realizado una serie de transformaciones;
- la introducción de restricciones geométricas a la hora de dibujar;
- la posibilidad de manejar estas restricciones a través de iconos;
- la posibilidad de realizar copias tanto de objetos como de dichas restricciones;
- diversas técnicas de dibujado a través del lápiz óptico;
- la distinción entre el sistema de coordenadas local del objeto a dibujar y del dibujo total;
- la realización de operaciones recursivas a dibujos estructurados jerárquicamente;

[16] Roger F. T. Reflections on the Revolution: The Transition from Analogue to Digital Representations of Space, 1958-1988. The American Cartographer: Journal of American Congress on Surveying and Mapping. 1988. Vol 15, n.3.

[17] Boehm, B. W., Tabular representation of multivariate functions with application to topographic modeling. Proceedings, 22nd National Conference, Association for Computing Machinery, 1967.pp. 403-415.

[18] Mark, D.M. The history of geographic information systems: invention and re-invention of Triangulated Irregular Networks (TINs). In Proceedings of ASPRS-RT I, 1998.Annual Conference. pp. 284–289.

[19] Chrisman, N. Charting the Unknown: How Computer Mapping at Harvard became GIS. ESRI Press: Redlands CA. 2006. ISBN: 978-1589481183.

[20] Sutherland, I. Sketchpad: A Man Machine Graphical Communication System. PhD thesis, MIT, 1963.

Figura 2.6.Ivan Sutherland manejando Sketchpad

Los avances del hardware pronto se mostraron insuficientes ante el gran consumo de recursos que implicaban las representaciones gráficas. El número de polígonos a representar se detectó muy pronto como el principal cuello de botella y en este mismo año Lawrence Roberts [21] presenta el primer algoritmo para la eliminación de superficies ocultas. La década de los 70 se encargaría de agilizar estos algoritmos.

En 1965 en su artículo "The Ultimate Display " [22], Sutherland plantea la primera descripción de lo que más tarde se llamaría Realidad Virtual . Dos de las aportaciones sin las cuales dicho término no podría existir tuvieron lugar en el MIT (Massachusetts Institute of Technology), gracias a Larry Roberts e Ivan Sutherland entre otros. La primera contribución implicó la investigación y desarrollos que permitieron al tubo de rayos catódicos servir como mecanismo de generación de imágenes por ordenador. La segunda consistió en el desarrollo de interfaces interactivas que permitieran al usuario interactuar con las imágenes generadas.

En esta década la generación de escenarios para los simuladores (hasta el momento solo aéreos y espaciales, como el *Gemini Mission Simulator*) se basaba en el empleo de cámaras de televisión, que colocadas sobre un montaje móvil, se desplazaban en función de las señales que le enviaba el ordenador con el fin de grabar diferentes partes de una maqueta o de grandes fotografías aéreas de la superficie terrestre (Figura 2.7). Mediante un sistema de proyección de tipo planetario se proyectaban estrellas, el horizonte y vehículos en los simuladores del Johnson Space Center.

Figura 2.7.Circuito de televisión cerrado sobre maqueta de terreno.

En 1966 Iván Sutherland construyó el primer dispositivo estereoscópico controlado por computadora y tres años más tarde, el Departamento de Defensa de los Estados Unidos financió el primer casco de visualización.

[21] Roberts, L.G. Machine Perception of Three-Dimensional Solids. In Proceedings of Outstanding Dissertations in the Computer Sciences 1963.

[22] Sutherland, I. E. The ultimate display. In Proceedings of IFIPS Congress. New York City, NY, 1965. Vol. 2, pp. 506–508.

La *Bell Helicopter Company* jugó un papel importante en el desarrollo de cascos virtuales. Éstos recibían la señal de una cámara de infrarrojos servo-controlada, montada en la parte baja del helicóptero. La cámara se movía según lo hacía la cabeza del piloto, de manera que el campo de visión del piloto era el mismo que el de la cámara. A través de éste sistema se pretendía entrenar a los pilotos en el aterrizaje nocturno sobre terrenos accidentados. Las experiencias vividas con dicho simulador demostraron la posibilidad del hombre de sentirse totalmente sumergido en un entorno remoto a través de los ojos de una cámara.

En Harvard, Sutherland y un estudiante, Robert Sproull convirtieron los sistemas de "realidad remota" de la Bell Helicopter en un proyecto de Realidad Virtual, sustituyendo para ello la cámara por imágenes generadas por computador. El primer entorno virtual consistió en una habitación en vista alámbrica, con los cuatro puntos cardinales sobre sus paredes. El visitante podía entrar en la habitación por la puerta este y girar para mirar a través de la ventana en las otras tres direcciones. De esta manera el casco virtual dio paso a la generación de los primeros entornos virtuales.

En la década de los 70, surgen los primeros sistemas de información geográfica comerciales: en 1973, MAGI, Maryland Automatic Geographic Information y en 1979 se desarrolla ODYSSEY GIS. Por otro lado en 1972 se lanza el primer satélite Landsat causando una revolución en el mundo de la percepción remota. En 1979 Tanaka desarrolló la que fue probablemente la primera transformación de perspectiva de una imagen tomada del mundo real: transformó una imagen Landsat del monte Fuji en una vista horizontal a baja altitud sobre un modelo 3d consistente en una malla regular de elevaciones extraída de un mapa de contornos.

Esta década se caracteriza por los importantes desarrollos software conseguidos. A medida que las exigencias de representación fueron mayores y con ellas los costes computacionales, el hardware disponible se hizo insuficiente. Era necesario el desarrollo de nuevas herramientas de optimización que agilizaran la carga gráfica. La universidad de Utah destacó por sus contribuciones en este ámbito. Entre sus aportaciones se encuentran algunos de los algoritmos más influyentes en la historia de la informática gráfica.

La aparición de las primeras superficies opacas, motivó el desarrollo de algoritmos para la determinación de cuáles de éstas eran ocultas. La no representación de dichas superficies ayudaba a disminuir la carga gráfica. En este campo destacaron las aportaciones de Romney, Warnock y Watkins. Sutherland realiza una comparativa sobre diversos algoritmos de eliminación de superficies ocultas [23].

Gracias a la agilización conseguida con motivo de la eliminación de las superficies ocultas, comenzó la búsqueda de un mayor grado de realismo:

- Aparecen las primeras superficies texturadas. Catmull, considerado el pionero en este campo, fue el primero en demostrar el mapeado de un patrón de textura en superficies planas y cilíndricas [24] . Posteriormente Blinn y Newell [25] dieron un paso más mapeando texturas fotográficas en superficies bicúbicas.
- El sombreado y la iluminación adquieren gran relevancia a la hora de dar realismo a la escena. Son fundamentales las aportaciones de Crow [26] , Phong [27], Gouraud [28] y Blinn [29].

[23] Sutherland, I. E, Sproull, R. F. and Schumacker, R. A. A Characterization of Ten Hidden-Surface Algorithms. In Proceedings of Computing Surveys 1974. Vol 6, n.1, pp 1-55.

[24] Blinn, J.F. Models of light reflection for computer synthesized pictures. In Proceedings of Computer Graphics 1997. Vol 11, pp 192-198.

[25] Blinn, J.F. and Newell M.E. Texture and Reflection in Computer Generated Images. In Proceedings of Commun. ACM 1979. Vol 19, n.10, pp. 542-547 .

[26] Crow, F. C. Shadow algorithms for computer graphics. In Proceedings of Commun. ACM 1977, pp.242-248.

[27] Phong, B. T. Illumination for computer generated pictures. In Proceedings of Commun. ACM 1975. Vol 18, n.6, pp 311-317.

- Aparecen nuevas técnicas para el modelado de superficies. La aproximación de superficies por polígonos planos es sustituida por ecuaciones matemáticas, obteniéndose superficies más realistas. Entre las diversas formulaciones matemáticas empleadas hay que destacar:
  - Modelado procedural (Newell) [30].
  - Splines (Lyche,Riesenfeld, Cohen) [31].
  - Beta-splines (Barsky) [32].

A su vez, se continúan investigando nuevas técnicas para aumentar la velocidad de renderizado. En 1976 James Clark [33], motivado por la búsqueda de un mayor realismo en los gráficos por computador y una mayor eficacia en los algoritmos que los generan, introduce una serie de conceptos claves sobre los que hoy en día siguen sustentándose los pilares de toda representación gráfica en tiempo real:

- considera que la precisión con la que un objeto es renderizado debe ser proporcional al número de píxeles que ocupa en pantalla. Propone así el empleo de múltiples niveles de detalle (level of detail, LOD) para la representación de entornos complejos. Para conseguir esta representación define una jerarquía en la que la raíz la constituye todo el escenario y de ella cuelgan todos los objetos contenidos en éste (Figura 2.8). El primer nodo representativo de cada objeto está formado por su representación más simplificada. A medida que descendemos en el árbol se obtienen versiones más detalladas de cada objeto. Clark también resalta la necesidad de tener en cuenta medidas perceptuales a la hora de establecer los criterios de selección del nivel de detalle, factores tales como la excentricidad (medida del grado de visibilidad que un objeto tiene dentro del campo periférico del usuario) y la velocidad de la cámara a la hora de determinar el nivel de detalle (a más velocidad, menos nivel de detalle es requerido).
- A partir de esta jerarquía propone un rápido algoritmo de recorte (clipping) que desciende recursivamente por el árbol comprobando el área ocupada por cada objeto (realmente, por su esfera de abarque) y determinando las zonas que entran dentro del campo de visión.
- Ante el exceso de memoria que requieren los grandes entornos virtuales sugiere la definición de un "grupo de trabajo", constituido por toda la información del entorno necesaria para conseguir que la visualización sea fluida, que sea almacenado en memoria de acceso inmediato.
- Propone para la determinación de las superficies visibles un algoritmo descendiente recursivo en el que para cada nivel todos los objetos son ordenados según su volumen de abarque. Si el test de visibilidad determina que un objeto es ocultado por otro, él y todos sus descendientes son eliminados y no se tienen en cuenta en los restantes cálculos.

---

[28] Gouraud, H. Continuous shading of curved surfaces. IEEE Transactions on Computers, 1971. Vol. 20, n. 6, pp 623–628.

[29] Blinn, J.F., Models of light reflection for computer synthesized pictures. In Proceedings of Computer Graphics 1997. Vol.11, pp 192-198.

[30] Newell, M. The Utilization of Procedure Models in Digital Image Synthesis, Ph.D. dissertation, Dept. of Computer Science, University of Utah, 1975.

[31] Cohen, E., Lyche, T. and Riesenfeld R. Discrete Box Splines and Refinement Algorithms. Computer Aided Geometric Design, 1984. Vol 1, n.2, pp 131-148.

[32] Barsky B. A. and Beatty, J. C. Local control of bias and tension in beta-spline. In Proceedings of Computer Graphics, 1983. Vol. 17, n. 3, pp. 193-218.

[33] Clark, J. H. Hierarchical geometric models for visible surface algorithms. Communications of the ACM, 1976. Vol.19, n.10, pp.547-554.

- Introduce el concepto de procesamiento en paralelo del grafo de la escena, que es la base que hoy en día sigue toda la unidad de procesamiento gráfico (Graphics Processing Unit, GPU).

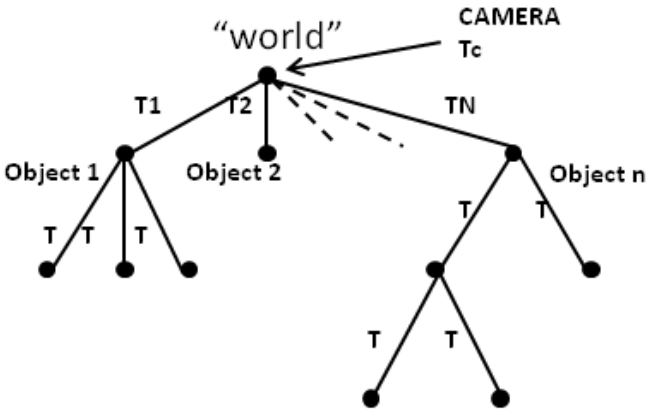


Figura 2.8.Grafo de la escena.

El primer sistema para la generación de imágenes por computador para simuladores fue desarrollado por General Electric para la NASA. Las imágenes para el simulador del Apollo y posteriormente del Shuttle y el Skylab se generarían por computador.

Los primeros simuladores de vuelo de esta nueva generación, se limitaban a representar escenas nocturnas en las que únicamente eran reconocibles las pistas de aterrizaje y despegue, correctamente iluminadas y señalizadas. Uno de estos primeros sistemas fue el realizado en 1971 por McDonnell-Douglas Electronics Corporation, llamado Vital II (Virtual Image Takeoff And Landing).

La segunda fase del Vital II permitió la visualización de luces direccionales y parpadeantes. La tercera fase permitió la representación de escenarios de día, con un mayor detalle: era posible el reconocimiento del aeropuerto, el terreno y las señales (Figura 2.9).

Figura 2.9.Ejemplos de representación de las pistas de aterrizaje y despegue del Vital.

Las fuerzas aéreas se interesaron también por la simulación a bajas altitudes, para lo que se necesitaba una representación del terreno a una mayor resolución. Su simulador aéreo (ASPT: Advanced Simulator for Pilot Training) instalado en 1974 en Arizona, fue uno de los primeros ejemplos de un display múltiple, que sirvió de modelo para los simuladores de la década de los 80 y 90. En éste, el campo de visión era dividido entre múltiples tubos de rayos catódicos (CRTs) que rodeaban al piloto, cada uno de los cuales era alimentado por una señal procedente de un

ordenador independiente, pero todos ellos sincronizados: alineando los bordes de cada dispositivo con sus adyacentes daba la sensación de una imagen continua.

La década de los 80 permitió pasar de los multimillonarios proyectos militares que empleaban los cascos virtuales diseñados por Sutherland al empleo de ordenadores personales para las representaciones virtuales. En 1984, Michael McGreevy creó para la NASA la primera workstation para la generación de entornos virtuales, el VIVED (Virtual Visual Environment Display system).

En esta década los desarrollos realizados en el campo de la simulación aérea se extendieron a diversos tipos de vehículos marítimos y terrestres, destacándose los aportaciones de Evans and Sutherland [34] . La mayor proximidad del conductor a su entorno en este tipo de simuladores obligó a la introducción de un mayor grado de detalle de la escena.

El texturado se convirtió en una herramienta indispensable a la hora de disminuir el número de polígonos de la escena y aumentar su realismo. Feibush et al.[35] emplearon esta técnica para renderizar texturas sintéticas en las fachadas de modelos 3d de casas. Feibush y Greenberg[36] desarrollaron un sistema de mapeado de texturas para ser empleado en diseño arquitectónico. Posteriormente Economy y Bunker [37] y Bunker et al. [38] introdujeron el concepto de "celdas de textura" para cubrir el terreno en los simuladores aéreos mientras que Dungan et al. [39] emplearon texturas extraídas a partir de fotografías aéreas. Estos últimos autores fueron de los primeros en emplear texturas con diferente resolución como método anti- aliasing. Unos años más tarde, los investigadores de General Electric (Economy y Bunker) y de Honeywell (Scott [40] , Ericsson et al.[41]) desarrollaron aplicaciones para el texturado de fotografías de árboles individuales sobre planos que giraban según el punto de vista (*billboards*) y sobre modelos tridimensionales de árboles (Figura 2.10). Comienza así el empleo del canal alfa, con el objetivo de generar zonas transparentes para simular las hojas de los árboles. Bunker et al utilizaron también esta técnica para la representación de nubes.

Geoff Gadner, de Grumman Data System, aportó un interesante método para la generación de escenarios de la naturaleza mediante el empleo de superficies cuádricas texturadas a través de una función que regulaba la intensidad del sombreado y el grado de traslucidez de las escenas [42] (Figura 2.11).

[34] http://www.es.com/ [Última consulta: 8 Enero 2013].

[35] Feibush, E. A., Levoy, M., Cook, R. L. Synthetic Texturing Using Digital Filters. In Proceedings of Computer Graphics, 1980. (SIGGRAPH '80 Proceedings). Vol. 14, n. 3, pp. 294-301.

[36] Feibush, E. and Greenberg, D.P. Texture rendering system for architectural design. Computer-Aided Design, 1980. Vol. 12, pp 67-71.

[37] Economy, R. and Bunker, M. Advanced video object simulation. In Proceedings of the National Aerospace and Electronics Conference 1984 .pp 1065–1071.

[38] Economy, R., Bunker, M., Harvey, J. Cell texture-Its impact on computer image generation. In Proceedings of the Sixth Interservice/ Industry Training Equipment Conference 1984. pp 149–155.

[39] Dungan, W., Stenger, A. and Sutty, G. Texture tile considerations for raster graphics. Computer Graphics 1978. (Proceedings of SIGGRAPH'78). Vol 12, n. 3, pp 130–134.

[40] Scott, W. B. Image system to unite computer video. Aviation Week Space Technol. 1983. Pp 70–73.

[41] Erickson, C., Fairchild, K. M and Marvel, O. Simulation gets a new look. Defense Electron. 1984. pp 76–87.

[42] Gardner G.Y. Simulation of Natural Scenes Using Textured Quadric Surfaces. In Proceedings of Computer Graphics, 1984. Vol.18, n.3, pp. 11-20.

Figura 2.10.General Electric:"Cell Texture"

Figura 2.11.Ejemplo de árboles, nubes y montes modelados con superficies cuádricas texturadas.

A comienzos de los 80 investigadores de Honeywell [43] desarrollaron una técnica híbrida para la generación de entornos virtuales que denominaron *Computer Generated Synthetisized Imagery (CGSI)*. Este método combinaba las mejores características de la técnica *Computer Graphics Imagery (CGI)* y de la técnica *Computer Synthetisized Imagery (CSI). La* técnica CGI, consistente en la generación por ordenador de modelos poligonales coloreados y sombreados, permitía una total libertad de movimientos dentro del entorno pero con un escaso grado de realismo. Por otro lado la CSI utilizaba como fondos fotografías digitales mientras que el resto de objetos 3d de la escena los realizaba por ordenador. De esta manera el grado de realismo era mayor sin embargo el movimiento estaba prácticamente limitado a una única dirección, únicamente se podía simular un limitado movimiento mediante el desplazamiento y zoom de la imagen de fondo. La nueva técnica CGSI combinaba lo mejor de ambas técnicas: la total libertad de movimientos del CGI y el realismo de la CSI. Esto se consiguió mediante la generación por ordenador del fondo y la inserción de fotografías para elementos de la escena como edificios, árboles..etc. Estos objetos eran fotografiados desde diferentes puntos de vista, digitalizados y almacenados en disco. En tiempo de ejecución cuando un determinado objeto era necesario se seleccionaba la vista más apropiada y se insertaba en el escenario generado por computador, con las correspondientes oclusiones y blending de bordes.

Mientras tanto, los investigadores de LTV Missiles and Electronics [44] (ahora Locheed-Martin-Vought Systems, LMVS) desarrollaron otra técnica híbrida a la que denominaron pseudo-película. Esta nueva técnica consistía en la generación de una imagen monocromática mediante la interpolación en tiempo de ejecución de un pequeño conjunto de diferentes fotografías oblicuas minuciosamente seleccionadas. En tiempo de preprocesado estas fotografías eran digitalizadas y almacenadas en disco. Una sistema mejorado de éste que incluía imágenes a color fue descrito por Hooks y Devarajan [45] y por Mundunuri y Hooks[46].

Unos años más tarde las fotografías oblicuas de la LTV serían sustituidas por mosaicos de fotografías aéreas [47] [48]. A su vez Honeywell sustituiría el fondo realizado por computador por la

[43] Baldwin, D. M., Goldiez, B. F., and Graf, C. P. A hybrid approach to high fidelity visual/sensor simulation. In Proceedings of the International Conference on Simulators, Conference Publication 226, Institution of Electrical Engineers, London and New York, 1983. Pp 1–6.

[44] Hooks, J. T. and Devarajan, V. Simulated FLIR imagery using computer animated photographic terrain views. In Proceedings of IMAGE Conference II, 1981.pp 25–34.

[45] Hooks, J. T. and Devarajan, V. Digital processing of color photography for visual simulations. In Proceedings of the Third Interservice/Industry Training Equipment and Exhibition, National Security Industrial Association.1981.

[46] Munduri, B. K. and Hooks, J. T. An algorithm for generation of super wide field of view scanned images. In Proceedings of the 38th Annual SPSE Conference 1985. (Springfield,VA, May 12–16).pp 154–158..

[47] Devarajan, V. and Chen, Y. P. Advanced data and picture transformation system. In Proceedings of Electronic Imaging '86. The Institute for Graphic Communications1986 (Boston, MA). pp 767–772.

generación en tiempo real de imágenes a partir de fotografías aéreas [49]. En ambos casos las fotografías eran en primer lugar ortorrectificadas y ensambladas para estar correlacionadas con el modelo de elevación del terreno, que era convertido en una malla de triángulos.

En esta década los investigadores de General Electric y de Evans and Sutherland continuaron con nuevas investigaciones. Con el fin de minimizar el número de polígonos a utilizar representaban el terreno mediante una malla irregular de triángulos (*Triangular Irregular Network*, TIN). Estas TINs eran generalmente realizadas mediante la triangulación de Delaunay [50] o mediante alguna técnica de subdivisión jerárquica [51] a partir de una red regular de elevaciones. El objetivo de estas mallas irregulares de triángulos era adaptar el tamaño del triángulo según la irregularidad del terreno, de manera que terrenos lisos pudiesen representarse con un mínimo número de triángulos de gran tamaño y terrenos abruptos con un gran número de triángulos de pequeño tamaño. Resúmenes de muchos de estos primeros sistemas para la generación de simuladores de vuelo se encuentran en Schachter y Ahuja[52], Schachter [53] y Yan [54].

A medida que avanzó tanto el hardware como las tecnologías de la computación gráfica el número de polígonos permitido en la escena fue aumentando y el sombreado plano fue sustituido por sombreado Guouraud y Pong. Según la demanda de realismo fue aumentando se fueron desarrollando nuevas técnicas para la generación y aplicación de texturas sintéticas y extraídas de fotografías. En Robinson y Zimmerman [55] se hace un resumen de estas técnicas.

A partir de la década de los 90, asentadas las bases para la generación de grandes entornos virtuales, los esfuerzos se centran en la automatización del tratamiento de la información de partida y en el aumento del grado de realismo de la representación [56]. Los grandes avances conseguidos en las técnicas de percepción remota incrementan significativamente el número de fuentes de información disponibles. La gran heterogeneidad de datos supone el tener que afrontar problemas de diversa índole: diferentes grados de resolución, sistemas de proyección y fechas de actualización entre otros, motivan la existencia de incongruencias y errores. La resolución manual de dichos problemas es una ardua tarea que exige una gran inversión de tiempo y es un foco de nuevos errores. Es necesario emplear herramientas que aumenten la automatización en la gestión de dicha información [57].

---

[48] Devarajan, V. Image processing in visual systems for flight simulation. In Proceedings of SPIE 1989 (Bellingham, WA). Vol. 1075, pp 208–216.

[49] Hopkins, R. R. and Cooper, M. B. Making clouds. In Proceedings of the Tenth Interservice/Industry Training Systems Conference, National Security Industrial Association 1988 (Washington, DC, Nov. 29–Dec. 1). pp 209–212.

[50] Lee, D. and Schachter, B. Two algorithms for constructing a Delaunay triangulation. In International Journal of Computer Information Sciences 1980. Vol 9, n. 3, pp 219–242.

[51] Bunker, W. and Heartz, R. Perspective display simulation of terrain. Tech. Rep. AFHRL-TR-76-39, Defense Technical Information Center, Cameron Station, Alexandria, VA. 1976.

[52] Schachter, B. J. and Ahuja, N. A history of visual flight simulation. Comput. Graph.World 1980. pp 16–31.

[53] Schachter, B. J. Computer image generation for flight simulation. IEEE Comput.Graph. Appl. 1981. Pp 29–63.

[54] Yan, J. K. Advances in computer-generated imagery for flight simulation. IEEE Comput. Graph. Appl. 1985. pp 37–51.

[55] Robinson, J. and Zimmerman, S. Exploiting texture in an integrated training environment. In Proceedings of the Seventh Interservice/Industry Training Systems Conference 1985 (Washington, DC, Nov. 19–21), pp 113–121.

[56] M.F. Polis, Giffor, S.J., McKeown, D,M. Automated the Construction of Large-Scale Virtual Worlds. In Proceedings of IEEE Computer, 1995. Vol. 28, n.7, pp 57-65.

[57] Haala, N. Combining multiple data sources for urban data acquisition. In Proceedings of Photogrammetric Week, Stuttgart, Germany, 1999. pp. 329–339.

A su vez, el volumen de información a representar aumenta vertiginosamente, lo que impulsa el desarrollo de multitud de nuevos algoritmos para la optimización de la carga gráfica de la representación virtual [58].

Paralelamente a todos estos desarrollos software el hardware no ha dejado de evolucionar. Los recientes avances conseguidos en el campo de las GPUs están causando una revolución en las técnicas de optimización empleadas hasta el momento. Por un lado, las elevadas velocidades de cálculo de estos coprocesadores están permitiendo manejar cantidades de información hasta hace unos años impensables. Por otro lado, la posibilidad de programar tanto a nivel de vértice como de píxel está facilitando un grado de control total de la representación [59].

En el siguiente apartado se particularizan las diversas áreas actualmente involucradas en la generación de grandes entornos virtuales al caso de los simuladores de conducción terrestre. Se estudian las soluciones aportadas por las principales entidades dedicadas al mundo de la simulación y se concluye con los simuladores del CITEF, Centro de Investigaciones Ferroviarias que ha empleado para la construcción de sus entornos virtuales las soluciones propuestas por esta Tesis.

2.4 GENERACIÓN DE GRANDES ENTORNOS VIRTUALES. APLICACIÓN A SIMULADORES DE CONDUCCIÓN TERRESTRE.

No se puede hablar de la existencia de una técnica maestra válida para la generación de cualquier entorno virtual. El éxito de la generación vendrá determinado por el equilibrio:

Tipo y finalidad de la aplicación virtual + ***Hardware disponible*** + ***Información inicial*** + ***Técnicas de optimización***

A continuación se particularizan cada uno de estos factores para el caso de los simuladores de conducción terrestre guiada bajo PC, tema abordado en esta Tesis.

2.4.1 TIPOS Y FINALIDADES.

En el caso de los simuladores de conducción terrestre, las finalidades pueden ser muy diversas. Entre ellas cabe citar:

- **Análisis del tráfico** [60]: la definición topológica y funcional de la red circulatoria se convierte en el punto clave para alcanzar el éxito en la simulación. La especificidad de los formatos requeridos para asegurar la correcta comunicación con el módulo de conducción, hacen preferible el desarrollo de herramientas ad hoc.

[58] Akenine-Möller, T., Haines, E. Chapter 14: Acceleration Algorithms. Real-Time Rendering. A.K. Peters Ltd. 2008. ISBN: 978-1568814247.

[59] Bailey, M. , Cunningham, S. Graphics Shaders: Theory and Practice, Second Edition. A K Peters/ CRC Press 2012. ISBN: 978-1568814346.

[60] http://www.aimsun.com/ [Última consulta: 8 Enero 2013].

- **Entretenimiento** [61]: la calidad visual es fundamental, se explota así la utilización de herramientas comerciales junto al empleo de infografistas.

- **Aprendizaje y entrenamiento** [62]: la necesidad por parte del conductor de reconocer situaciones reales obliga a la generación de entornos realistas en los que la posibilidad de reestructuraciones es elevada. Se trata así de un caso idóneo para la generación de herramientas específicas tanto para la gestión de la información de partida como del modelado 3d del entorno.

- **Análisis de la respuesta humana ante diversos factores** [63]: Con este tipo de simuladores se pretende mejorar el conocimiento sobre la respuesta humana en la conducción ante diversos factores externos. El entorno virtual en la simulación no tiene como objetivo el disfrute del usuario, como puede ocurrir en un videojuego, ni la identificación de un determinado trazado, como puede ocurrir en un labor de entrenamiento ferroviario. En este caso, el entorno es un mero instrumento para situar al conductor en una determinada situación y analizar su respuesta, siendo los recursos empleados en la generación de dichos escenarios los imprescindibles para llevar a cabo estos tests. Se representan entornos genéricos, que no han de reproducir con fidelidad ningún trazado en particular, por lo que es innecesario el desarrollo de herramientas ad hoc para la creación de los mismos. El empleo de herramientas comerciales o de libre distribución cumplen los objetivos buscados.

La presente Tesis se centrará en los simuladores de conducción terrestre destinados al aprendizaje y entrenamiento. Las características y requisitos de los mismos serán evaluados en el Capítulo 3, en lo que esta Tesis ha llamado, *primera fase de construcción* de un entorno virtual.

2.4.2 HARDWARE.

A la hora de evaluar el hardware existe toda una gama de simuladores en función del coste como muestra la siguiente figura.

Figura 2.12.Simuladores ferroviarios en función del coste.

[61] http://www.railsimulator.com/ [Última consulta: 8 Enero 2013].

[62] http://www.corys.com/The-Tools---443.html [Última consulta: 8 Enero 2013].

[63] http://www.humanfirst.umn.edu/. [Última consulta: 8 Enero 2013].

En función del hardware empleado es posible distinguir cuatro tipos de entrenamiento, que se diferencian en el grado de inmersividad alcanzado por el alumno y los objetivos buscados (Figura 2.13):

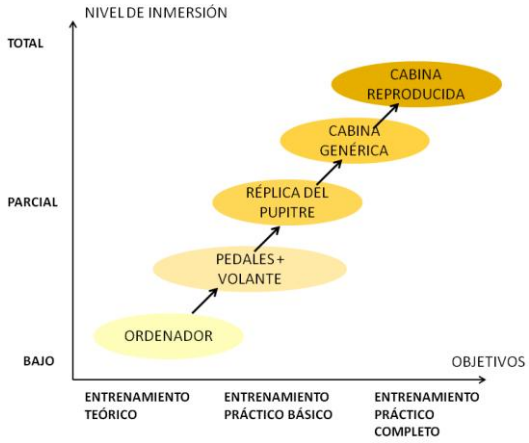


Figura 2.13.Tipos de entrenamiento.

1. El instructor dirige el entrenamiento en una clase apoyándose para ello en libros y aplicaciones multimedia (*Computer Assisted Instruction, CAI*)
2. Cada alumno dirige su propio entrenamiento con la ayuda de un ordenador (*Computer Based Training, CBT*).
3. El entrenamiento se lleva a cabo en ordenadores personales y se pueden distinguir dos tipos (Part Task Trainer, PTT):
    - Simuladores DeskTop**:** suelen llevar asociados pedales y volante. Estos simuladores son los de mayor difusión en el ámbito del aprendizaje a la conducción (autoescuelas), médico y académico. Algunos ejemplos de este tipo de simuladores los encontramos en STISIM [64] (Systems Technology, Inc), NADS [65] y en XPISimulation [66].

Figura 2.14.STISIM

Figura 2.15.XPI Simulation

[64] http://www.stisimdrive.com/. [Última consulta: 8 Enero 2013].

[65] http://www.nads-sc.uiowa.edu/sim_nads2.php. [Última consulta: 8 Enero 2013]

[66] http://www.xpisimulation.com/products/xp100.htm. [Última consulta: 8 Enero 2013]

- Simuladores de pupitre replicado: Incorporan además de lo visto para los simuladores DeskTop una réplica del pupitre de conducción, como muestra la siguiente figura.

Figura 2.16.Desk Simulator de CASSIDIAN (EADS).

4. Finalmente el entrenamiento con cabina, que simula de la manera más fidedigna el modo de operar en condiciones reales (Full Mission Simulator, FMS). En este grupo se pueden distinguir dos categorías:

   - Simuladores de cabina genérica: generalmente incluyen una cabina y pantallas de proyección conectadas a un ordenador personal. Estos simuladores están creciendo en popularidad tanto en el ámbito académico como gubernamental debido a las experiencias tan realistas que son capaces de reproducir. Ejemplos de este de simuladores son XPISimulation [67], FAAC [68] y NADS [69].

Figura 2.17.XPISIMULATION

Figura 2.18.NADS

   - Simuladores de cabina de alto grado de fidelidad: son los más avanzados y sofisticados. Además de los elementos incorporados en un simulador de coste medio, suelen disponer de una plataforma *Stewart* o *hexapod* que permite el movimiento. Estos simuladores requieren de un hardware, software y componentes estructurales altamente sofisticados. Debido a los elevados costes asociados a este tipo de simuladores, estos solo se encuentran disponibles en universidades, agencias gubernamentales y en los centros de investigación de los grandes fabricantes de automóviles. Ejemplos de este tipo de simuladores son NADS [70], EADS [71], RENAULT [72], CITEF [73], Kraus-Maffei (KMW) [74].

[67] http://www.xpisimulation.com/products/ds2.htm [Última consulta: 8 Enero 2013].

[68] http://www.faac.com/ [Última consulta: 8 Enero 2013].

[69] http://www.nads-sc.uiowa.edu/sim_minisim.php. [Última consulta: 8 Enero 2013]

[70] http://www.nads-sc.uiowa.edu/sim_nads1.php . [Última consulta: 8 Enero 2013].

[71] http://www.cassidian.com [Última consulta: 8 Enero 2013].

Figura 2.19.Renault.

Figura 2.20.CASSIDIAN (EADS).

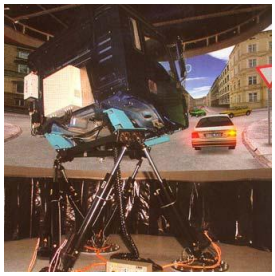


Figura 2.21.KMW.

Figura 2.22.National Advance Driving Simulator.

La Tecnología Modular pretende ser una metodología constructiva válida para todos los simuladores desarrollados por el CITEF, independientemente del grado de inmersividad y costes implicados. Estos entornos se generan en un PC, cuyas prestaciones definirán el valor óptimo de diversos parámetros que la Tecnología Modular permite configurar con el fin de proporcionar siempre la solución más óptima, según se verá en el Capítulo 5.

2.4.3 INFORMACIÓN INICIAL.

En cuanto a la información de partida ésta podrá ser real o ficticia. En cualquier caso, existen numerosas herramientas comerciales y de libre distribución dedicadas a su procesamiento y generación [75]. Sin embargo, el hecho de que no estén diseñadas para solventar los objetivos específicos de un determinado proyecto, supone asumir una serie de riesgos:

- Dichas herramientas en ocasiones emplean formatos propios, escasamente documentados, lo que implica:
    - Difícil manejo de la información resultante. Esto complica la introducción de futuras reestructuraciones del entorno, así como su optimización desde el punto de vista de la carga gráfica y circulación a su través.
    - La necesidad de convertir dichos formatos a los empleados por el motor gráfico, lo que generalmente dista mucho de ser una tarea trivial.
- No incorporación de una definición topológica que permita la circulación a través del

[72] http://www.experts.renault.com/ [Última consulta: 8 Enero 2013].

[73] http://www.citef.etsii.upm.es [Última consulta: 8 Enero 2013].

[74] http://www.kmweg.de [Última consulta: 8 Enero 2013].

[75] http://www.vterrain.org/Packages/Com [Última consulta: 8 Enero 2013]

entorno y en caso de incluirla la posible incompatibilidad con los protocolos establecidos por el módulo de conducción.

- Difícil y en ocasiones inexistente control de las deformaciones locales y globales que sufren los datos como consecuencia de las discrepancias entre las fuentes de información. En determinadas aplicaciones, como las simulaciones de conducción, las deformaciones admisibles son muy restrictivas y estos errores pueden ser excesivos.

- Dependencia de las actualizaciones de la aplicación, que en el peor de los casos pueden llegar a desaparecer.

- Incremento de los costes en el caso de optar por herramientas comerciales.

Esta Tesis resuelve la problemática asociada a este apartado en la que define como *segunda fase de construcción* de un entorno virtual, desarrollando para ello sus propios algoritmos y herramientas según será explicado en el Capítulo 4.

2.4.4 TÉCNICAS DE OPTIMIZACIÓN.

Las técnicas de optimización se basan en el control de las características de la escena teniendo en cuenta la arquitectura del sistema gráfico y sus limitaciones en la transferencia de datos. Su objetivo es una representación eficiente de la escena y la liberación de recursos para otros aspectos de la representación, como pueden ser los comportamientos dinámicos de objetos.

Estas técnicas son compartidas por todos los sistemas de visualización en tiempo real, aunque dependiendo de las características de cada sistema en concreto se aplicarán de una forma u otra. Existen en el mercado numerosos sistemas de visualización que permiten la representación de entornos virtuales en tiempo real, sin embargo, presentan el inconveniente de no poder conectarse a módulos de comportamiento o en caso de permitirlo, suele ser a través de interfaces complejas y de escasa flexibilidad. Con el fin de solventar estas limitaciones, los simuladores desarrollados por el CITEF usan un motor gráfico propio, que será el empleado por esta Tesis para la visualización de sus entornos.

Esta Tesis afronta la problemática asociada a este apartado en las que define como *primera y tercera fase de construcción* de un entorno virtual, según será explicado en los Capítulos 3 y 5.

Como conclusión se puede decir, que a medida que aumentan las necesidades de realismo y la envergadura del proyecto, aumenta la necesidad de automatización [76] y la tendencia es hacia el desarrollo de aplicaciones personalizadas que únicamente empleen ayuda externa a través del uso de librerías especializadas. De esta manera se consigue un control total de la generación del entorno, permitiendo la introducción de modificaciones y desapareciendo las posibles desavenencias con la optimización gráfica (subsistema motor gráfico) y la circulación a través del entorno (subsistema de simulación). Esta programación específica podrá ser llevada a cabo por el equipo responsable de la simulación o ser subcontratada a otras empresas, como se comentará en el próximo apartado.

A continuación se comentan las aportaciones más revelantes que las principales entidades dedicadas al mundo de la simulación de conducción están actualmente aportando en la generación de grandes entornos virtuales.

[76] Buchholz, H., Döllner, J., Ross, L., Kleinschmit, B. Automated Construction of Urban Terrain Models. Proceedings of the 12th International Symposium on Spatial Data Handling (SDH 2006), Springer-Verlag, 2006. pp 547-562.

2.5 ENTIDADES MÁS DESTACADAS EN LA GENERACIÓN DE ENTORNOS VIRTUALES PARA SIMULADORES DE CONDUCCIÓN TERRESTRE.

En los siguientes apartados se indicarán algunas de las entidades más destacadas por su aportación en el área de generación del entorno virtual en el mundo de la simulación de conducción terrestre. En primer lugar se enumerarán las principales empresas proveedoras de COTS (*) y se describirán la gama de productos desarrollados por cada una de ellas. A continuación se expondrán las soluciones que han adoptado para la generación de sus entornos virtuales entidades destacadas en el mundo de la simulación de conducción terrestre. Finalmente se concluirá con los simuladores del CITEF.

2.5.1 EMPRESAS PROVEEDORAS DE COTS.

OKTAL:

Oktal [77] ofrece tanto soluciones prototipo, como soluciones específicas particularizadas a los requisitos del cliente. Bien por cuenta propia o en asociación (principalmente con RENAULT, PSA Peugeot Citroen e INRETS), provee simuladores de conducción de coches y camiones desarrollados total o parcialmente a través de su paquete de software SCANeR^{TM}studio [78].

SCANeR^{TM}studio ofrece herramientas para la simulación que van desde la generación de bases de datos tridimensionales hasta la simulación interactiva en tiempo real y post-procesado. Estructurado en diversos módulos: Terreno,Vehículo, Escenario, Simulación y Análisis, permite crear toda la información necesaria para el simulador, incluida la información del tráfico y dinámica vehicular, que codifica en un archivo de configuración con su propio formato.

Este software está siendo utilizado por gran cantidad de simuladores de conducción a lo largo del mundo: Audi (Alemania), DECOMA (Canadá), PSA Peugeot Citroen (Francia), RENAULT(Francia), VOLVO (Francia), Universidad de Minesota (USA), Universidad de Nápoles, SINTEF (Noruega), TRL (UK), entre otras.

Figura 2.23.Ejemplos de entornos generados con la ayuda de SCANeR^{TM}Studio.

Para sus proyectos de simulación ferroviaria posee su software OKSimRail.

(*) COTS, Commercial-Off-The-Shelf: Software o Hardware desarrollado por terceros.

[77] http://www.oktal.fr [Última consulta: 8 Enero 2013].

[78] http://www.scanersimulation.com [Última consulta: 8 Enero 2013].

PRESAGIS:

Presagis [79] fue fundada en 2007 por CAE Inc. [80] mediante la adquisición de algunas de las empresas que colaboraban de una manera más activa en la generación de entornos virtuales del momento: Engenuity Technologies, MultiGen-Paradigm y TERREX. Productos altamente difundidos como Creator, STAGE, Terra Vista, VAPS y Vega Prime habían sido legado de estas empresas.

Presagis combina el empleo de componentes COTS (*Commercial-off-the-Shelf*) con desarrollos propios, particularizados en caso necesario, a los requisitos del cliente. A través de su paquete de software STAGE [81], ofrece soluciones que abarcan todo el proceso constructivo de la simulación.

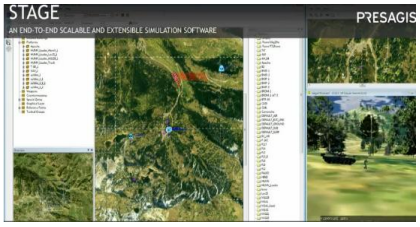


Figura 2.24.Software STAGE.

En concreto, su herramienta STAGE Scenario permite la construcción de entornos visuales y dispone de:

- un editor de base de datos, que permite su creación y apertura;
- un editor de escenarios, a través del cuál se pueden visualizar y colocar determinados elementos;
- un visualizador que permite la revisión continuada del entorno generado;
- un editor de scripts para la programación de comportamientos de determinados objetos del escenario;
- un editor para la generación de entornos ficticios.

Entre los clientes de Presagis se encuentran Saab, Kraus-Maffei Wegmann, Thales, Volkswagen, Honda , DaimlerChrysle, Lockheed Martin , Airbus, BAE SYSTEMS , Sony Computer Entertainment America (SCEA), The Boeing Company Vivendi Games y Midway Games entre otros.

QUANTUM3D.

Quantum3D[82] combina el empleo de componentes COTS (*Commercial-off-the-Shelf*) con desarrollos propios, particularizados, en caso necesario, a los requisitos del cliente.

[79] http://www.presagis.com/ [Última consulta: 8 Enero 2013]

[80] http://www.cae.com/ [Última consulta: 8 Enero 2013]

[81] http://www.presagis.com/products_services/products/modeling-simulation/simulation/stage/ . [Última consulta: 8 Enero 2013]

[82] http://www.quantum3d.com/. [Última consulta: 8 Enero 2013]

Como productos propios para la creación de entornos virtuales en la simulación de conducción terrestre destacan Lidar Fusion Tool Suite [83], CatalystSE [84], GeoScapeSE [85] y Facets [86]. Se trata de soluciones integradas basadas en el empleo de herramientas COTS que permiten al cliente el desarrollo, modificación y mantenimiento de grandes bases de datos geospecíficas. Entre las herramientas COTS empleadas se encuentran:

- ESRI ArcView [87], para la creación y edición de datos geospecíficos.
- TERREX TerraVista Pro Builder[88], para la creación, modificación y optimización del terreno en formato TerraPage y OpenFlight.
- MultiGen-Paradigm Creador ™ [89] , para la creación, modificación y optimización de modelos 3d en formato OpenFlight.
- Photoshop [90], para el tratamiento de texturas.
- Mantis [91], permite sobrevolar el entorno generado en tiempo real para validarlo.

Quantum3D es empleado en multitud de simuladores, como NADS, EWS (English Welsh & Scottish Railway), Lockheed Martin Overseas Corporation, Indra Systemas, Alenia Marconi, Thales Training and Simulation y Cory/TESS y por Ford en su simulador VIRTTEX, mostrado en la figura siguiente.

Figura 2.25.Imágenes tomadas del simulador VIRTTEX de Ford.

[83] http://www.quantum3d.com/solutions/real-time/synthetic/lidar_fusion_tool_suite.html . [Última consulta: 8 Enero 2013]

[84] http://www.quantum3d.com/solutions/real-time/synthetic/catalyst_dbgs_6000.html . [Última consulta: 8 Enero 2013]

[85] http://www.quantum3d.com/solutions/real-time/synthetic/geoscape/geoscape_se.html . [Última consulta: 8 Enero 2013]

[86] http://www.quantum3d.com/solutions/real-time/synthetic/facets.html . [Última consulta: 8 Enero 2013]

[87] http://www.esri.com/software/arcgis/ [Última consulta: 8 Enero 2013]

[88] http://www.presagis.com/products_services/products/modeling-simulation/content_creation/terra_vista/ [Última consulta: 8 Enero 2013]

[89] http://www.presagis.com/products_services/products/modeling-simulation/content_creation/creator/ [Última consulta: 8 Enero 2013]

[90] http://www.adobe.com/es/products/photoshopfamily.html [Última consulta:: 8 Enero 2013].

[91] http://www.quantum3d.com/solutions/real-time/software/mantis.html [Última consulta: 8 Enero 2013]

2.5.2 ENTIDADES DESTACADAS DEDICADAS A LA GENERACIÓN DE SIMULADORES.

NADS:

El National Advance Driving Simulator [92], NADS, desarrollado por la National Highway Traffic Safety Administration (NHTSA) y situado en la Universidad de Iowa, se considera el simulador de conducción más avanzado del mundo. La universidad es responsable de llevar a cabo todas sus labores de investigación, mantenimiento y actualización.

NADS ha desarrollado diversas herramientas de software para la generación de entornos virtuales [93] [94] [95]. Estas herramientas no solo están orientadas hacia el modelado estático de la escena, sino que permiten regular determinados factores relacionados con el tráfico y señalización.

El *Tile Mosaic Tool* [96], es un editor para la generación estática de la escena que permite al usuario la creación de un entorno virtual mediante la colocación en una matriz de celdas, de una serie de losetas prefabricadas (*tiles*). Estas losetas llevan incorporadas carreteras, edificios, vegetación y demás elementos característicos de un entorno. Sus bordes responden a una serie de perfiles predefinidos con el objetivo de producir un correcto ajuste entre ellos. El editor se encarga de verificar dicha continuidad en las fronteras de cada loseta y de garantizar la coherencia de conexiones entre carreteras. Cada loseta esta vinculada a tres bases de datos correlacionadas:

- Una base de datos visual, que representa el modelado 3d de la escena con diversos niveles de detalle (lods).
- Una base de datos que controla la geometría y topología de la red de carreteras del escenario.
- Una base de datos del terreno.

El editor se encarga de mantener la coherencia entre estas tres bases de datos. Por otro lado el *Interactive Scenario Authoring Tool* (ISAT) se encarga de regular determinados factores relacionados con el tráfico y señalización así como el comportamiento de determinados elementos del escenario (Figura 2.26).

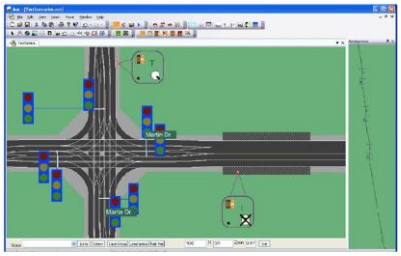


Figura 2.26.Interactive Scenario Authoring Tool (ISAT).

[92] http://www.nads-sc.uiowa.edu/ [Última consulta: 8 Enero 2013]

[93] Wu, Q., Oza, A., and Mourant, R. R. Pedestrian Scenario Design and Performance Assessment in Driving Simulations, Accepted for publication. In Proceedings of the Driving Simulation Conference North America, Orlando, Florida. 2005.

[94] Suresh, P. and Mourant , R. R, A Tile Manager for Deploying Scenarios in Virtual Driving Environments. Proceedings of the Driving Simulation Conference North America, December, 2005.

[95] Papelis, Y., Ahmad, O., and Watson, G. Scenario Definition and Control for the National Advanced Driving Simulator. In Proceedings of the Driving Simulation Conference North America, Dearborn, Michigan. 2003.

[96] http://www.nads-sc.uiowa.edu/media/pdf/NADS_ISAT.pdf . [Última consulta: 8 Enero 2013]

CASSIDIAN

Cassidian [97], conocida hasta el 17 de Septiembre del 2010 como EADS Security & Defense, es uno de los líderes en simuladores de conducción del mercado. Su programa "Simulation and Training" ofrece un sistema de entrenamiento para todo tipo de vehículos terrestres y está siendo utilizado por gran cantidad de simuladores en Europa. Dispone de toda una gama de productos que van desde simuladores bajo PC hasta simuladores completamente equipados en plataformas con 6 grados de libertad. Para la creación de sus entornos virtuales emplea herramientas COTS y aplicaciones desarrolladas ad hoc para cada proyecto. Un ejemplo es la herramienta desarrollada para TRENITALIA que muestra la siguiente figura.

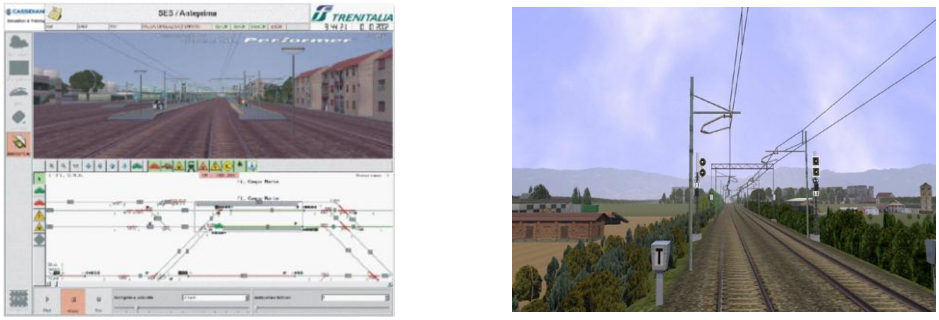


Figura 2.27.Ejemplos de entornos virtuales generados por Cassidian y su herramienta para la generación de los mismos.

Los simuladores y los sistemas del entrenamiento desarrollados por Cassidian están en servicio en Alemania, Francia, Países Bajos, Italia, Reino Unido, Turquía y Medio Oriente. En 2011 desarrolló un simulador de conducción para el metro de Bangalore (Bangalore Metro Rail Corporation Limited, BMRCL), para el que ha construido 42km de línea metropolitana y 41 estaciones.

Figura 2.28.Simulador desarrollado por CASIDIAN para el metro de Bangalore.

XPISIMULATION

XPI Simulation [98] diseña y fabrica tecnología para simuladores de conducción de diverso tipo: vuelo, coches y trenes. Ofrece una gama de productos ajustables según el presupuesto y requisitos de funcionamiento. Desarrolla bases de datos visuales tanto reales como ficticias. Para ello emplea herramientas COTS como Creator, TerraVista (Presagis) y AutoCad. Para facilitar la

[97] http://www.cassidian.com/ . [Última consulta: 8 Enero 2013]

[98] http://www.xpisimulation.com/. [Última consulta: 8 Enero 2013]

generación de bases de datos reales ha creado su herramienta Trackgen [99], con la cuál se han llevado a cabo proyectos como el desarrollado junto a Dynalantic Corps para la New York City Transit Authority, en el que ha generado 250 millas del metro de Nueva York.

Entre los proyectos desarrollados se encuentra el metro de Taipei [100], contratado por la compañía de simulación de Taiwan, AIDC.

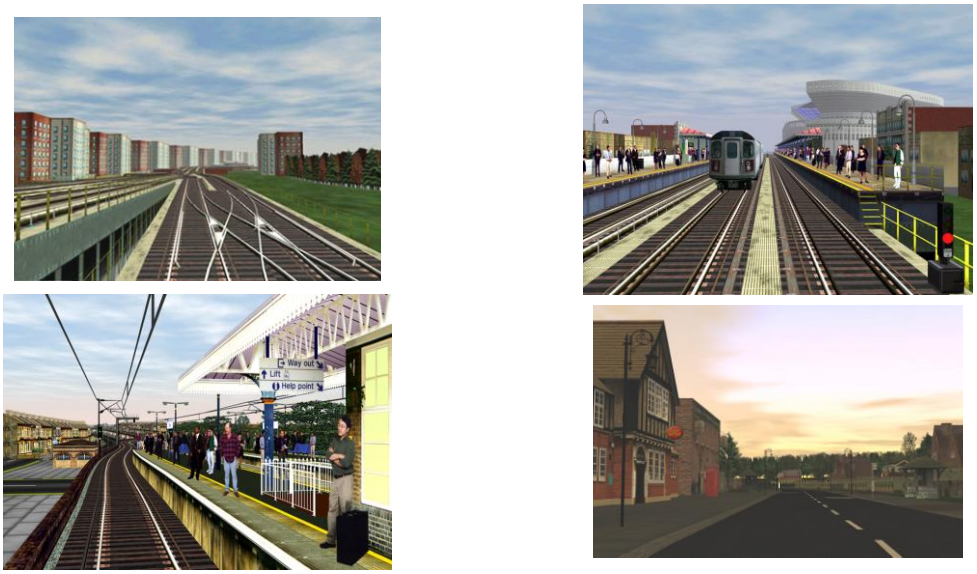


Figura 2.29.Ejemplos de diversos entornos virtuales desarrollados por XPI SIMULATION.

CORYS

CORYS T.E.S.S. [101], situada en Grenoble, lleva trabajando en el mundo de la simulación ferroviaria desde 1990, habiendo trabajado con numerosas compañías tanto en Europa, Asia, Australia, Africa y América del Norte.

Corys ha integrado todas sus tecnologías desarrolladas para simulación ferroviaria en el paquete de software Corail. Entre las herramientas creadas se encuentra un generador de trayectorias, *Track Builder Tool* [102] (Figura 2.30). Para la representación de entornos reales ha desarrollado sus propios algoritmos de extracción de la información (*GenAuto* [103]) que serán fuente de partida del *Track Builder Tool*.

CORYS T.E.S.S. ha subcontratado a Quantum3d en el desarrollo de determinados proyectos en Gran Bretaña (Virgin Trains), Australia y Singapur y ha llevado a cabo multitud de proyectos a nivel inernacional, entre ellos el metro de Londres y de Delhi [104].

[99] http://www.xpisimulation.com/products/database.htm. [Última consulta: 8 Enero 2013]

[100] http://www.xpisimulation.com/news/Sept_2006_01.htm. [Última consulta: 8 Enero 2013]

[101] http://www.corys.com/. [Última consulta: 8 Enero 2013]

[102] http://www.corys.com/The-Tools---443.html. [Última consulta: 8 Enero 2013]

[103] http://www.corys.com/GENAUTO--498.html [Última consulta: 8 Enero 2013]

[104] http://www.corys.com/Delhi-metro-a-second-simulator--1109.html. [Última consulta: 8 Enero 2013]

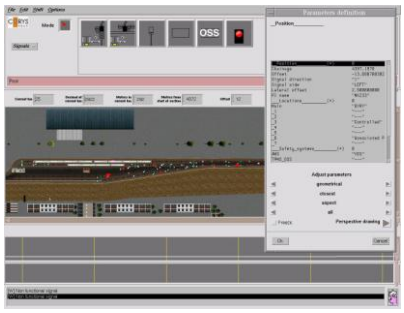


Figura 2.30.Track Builder Tool

Figura 2.31.Imagen del simulador del metro Londres llevado a cabo por CORYS.

KRAUS-MAFFEI WEGMANN (KMW)

Con sede en Munich y Kassel, la división de entrenamiento y simulación (Training & Simulation ,TS) de KMW [105] realiza toda una variedad de simuladores de conducción terrestre: coches, camiones, trenes y vehículos militares. La mayoría de estos simuladores se crean para el entrenamiento, aunque también en ocasiones para la investigación y el entretenimiento.

Desde el punto de vista de la generación del escenario virtual ofrecen una serie de editores, como CARMEN Track (Figura 2.33), que permiten la generación de bases de datos ficticias y la edición de entornos ya generados.

Entre sus clientes se encuentran Deutsche Bahn AG, Schweizerische Bundesbahn (SBB CFF FFS), Stadtwerke München (SWM), Metropolitano de Lisboa (ML), S-Bahn Berlin GMBH, Stuttgarter Straßenbahnen AG (SSB), Jakarta Railways (JABOTABEK), Projekt: SMC (Shanghai Metro), CFL Luxembourg, South West Trains Ltd., Siemens - Metro-Simulator für die Metro Oslo, Raja Passenger Trains / Iran, y National Express Group en UK entre otros.

Figura 2.32.Ejemplos de entornos virtuales desarrollados por KMW.

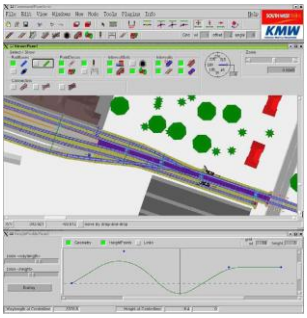


Figura 2.33.CARMEN-Track builder tool.

BENTLEY SYSTEMS

La compañía estadounidense Bentley Systems [106], empresa líder en generación de software para proyectos ferroviarios, de transporte y de construcción a gran escala, además de centrarse en el diseño y planificación de infraestructuras, desarrolla simuladores de entrenamiento.

[105] http://www.kmweg.de/gb/index.php / [Última consulta: 8 Enero 2013]

[106] http://www.bentley.com/en-US/ . [Última consulta: 8 Enero 2013]

Presenta una amplia gama de productos desarrollados por ellos mismos. Desde el punto de vista de la generación del entorno virtual destaca su variedad de herramientas SIG [107]. Entre sus proyectos desarrollados se encuentra el simulador de tren realizado para *First Great Western*.

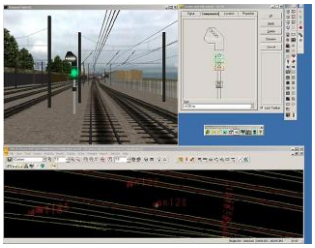


Figura 2.34.Herramienta para la configuración semafórica desarrollada por Bentley Systems.

Figura 2.35.Ejemplo de entorno virtual desarrollado por Bentley Systems.

FAAC

La empresa FAAC [108] suministra simuladores de autobuses camiones y trenes. Entre sus herramientas para la generación de escenarios virtuales destaca, *Scenario Tool Box* [109]. *Scenario Tool Box* está creado con el objetivo de permitir al instructor que esté dirigiendo el entrenamiento diseñar cómodamente escenarios ficticios e introducir diversas incidencias (Figura 2.36) .

Los proyectos para los que FAAC ha sido contratado son muy numerosos [110]. EL NYCY (New York City Transit) tras realizar un estudio sobre conductores con un año de prácticas en el simulador, asegura haber tenido lugar una reducción del 43% los accidentes. El Metro de Houston asegura una reducción del 28% de accidentes, Rochester, NY del 34% y Texas Association of Countries del 18%.

Figura 2.36.Ejemplos de entornos virtuales desarrollados por FAAC.

WIVW:

Wuerzburg Institute for Traffic Sciences, WIVW [111], con alrededor de 10 años de experiencia en la simulación de conducción, dispone de diversos simuladores: dos fijos, dos con plataforma móvil y diversos simuladores de bajo coste. Todos ellos están basados en su software

[107] http://www.bentley.com/en-US/Products/. [Última consulta: 8 Enero 2013]

[108] http://www.faac.com/. [Última consulta: 8 Enero 2013]

[109] http://www.faac.com/policesimulators.htm [Última consulta: 8 Enero 2013]

[110] http://www.faac.com/funding.html [Última consulta: 8 Enero 2013]

[111] http://www.wivw.de/. [Última consulta: 8 Enero 2013]

SILAB [112] (Figura 2.37). SILAB permite la generación de diversos tipos de escenarios, urbanos (Figura 2.38), interurbanos y rurales e incluye el comportamiento de vehículos y peatones. Entre los clientes de SILAB se encuentran BOSCH, IZVW y SIEMENS.

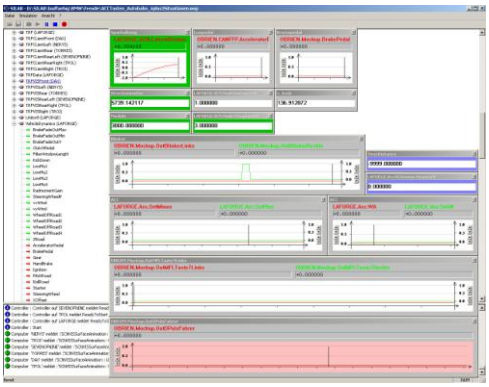


Figura 2.37.SILAB.

Figura 2.38.Ejemplo de entornos virtuales.

2.6 EL SIMULADOR DE CONDUCCIÓN TERRESTRE DE CITEF.

El Centro de Investigación en Tecnologías Ferroviarias, CITEF [113], nace dentro de la Escuela Técnica Superior de Ingenieros Industriales de la Universidad Politécnica de Madrid, en Noviembre de 1998 con el objeto de aunar esfuerzos conducentes a incrementar el desarrollo tecnológico en el sector del transporte por superficie, abrir nuevas líneas de investigación, participar en proyectos nacionales e internacionales y por último estimular la formación de Ingenieros Industriales para el transporte por superficie en general y para el ferrocarril en particular.

Así, desde su creación, CITEF organiza y desarrolla diferentes líneas de investigación que satisfacen la demanda tecnológica actual del transporte por superficie, tanto por los requisitos de interoperabilidad como por las directrices europeas al respecto. Para ello particulariza los esfuerzos cuando así son requeridos para el cumplimiento de los requerimientos propios de las administraciones, compañías y empresas del ferrocarril demandantes de un diseño cuyas restricciones vienen impuestas por la normativa EUROPEA en los diferentes grupos de trabajo, especialmente ferroviarios.

Una de las líneas de investigación llevadas a cabo por CITEF es la Simulación para Formación, siendo en este sector centro de referencia en España. En el entorno de los transportes por superficie, el empleo de simuladores está plenamente justificado debido a varios factores: el elevado coste tanto de las infraestructuras (públicas o privadas como circuitos de prueba) como de los vehículos, lo cual dificulta su inmovilización necesaria para la formación; la dificultad de generar situaciones de averías e incidencias para mostrar su correcta resolución al alumno; la imposibilidad en algunos casos de crear situaciones de riesgo para enseñar al alumno el procedimiento a seguir; etc. Destacan los simuladores desarrollados para la Empresa Municipal de Transportes de Madrid (EMT), Transports Metropolitans de Barcelona (TMB), Metro de Madrid y RENFE. Todos ellos consisten en la modelización de distintos tipos de vehículos con simulación de averías (eléctricas, neumáticas, etc.), con un alto grado de realismo.

[112] http://www.wivw.de/ProdukteDienstleistungen/SILAB/index.php.en [Última consulta: 8 Enero 2013]

[113] http://www.citef.etsii.upm.es [Última consulta: 8 Enero 2013].

Figura 2.39.Plataforma CITEF.

Figura 2.40.Instalaciones de CITEF para INVENSYS.

Figura 2.41.Instalaciones de CITEF en Metro de Madrid

En particular en el simulador de EMT, CITEF ha desarrollado un novedoso módulo de conducción de tráfico urbano con comportamiento aleatorio de los conductores que supone un avance significativo en el campo, al poderse utilizar en gran cantidad de simuladores con unas mínimas modificaciones: Metros Ligeros, Autobuses Interurbanos, Vehículos de transporte por carretera, etc. Para la creación de los entornos virtuales destinados a sus simulaciones de conducción ferroviaria y urbana CITEF ha empleado con gran éxito la tecnología desarrollada por esta Tesis, la Tecnología Modular, que se presenta a lo largo de los siguientes capítulos.

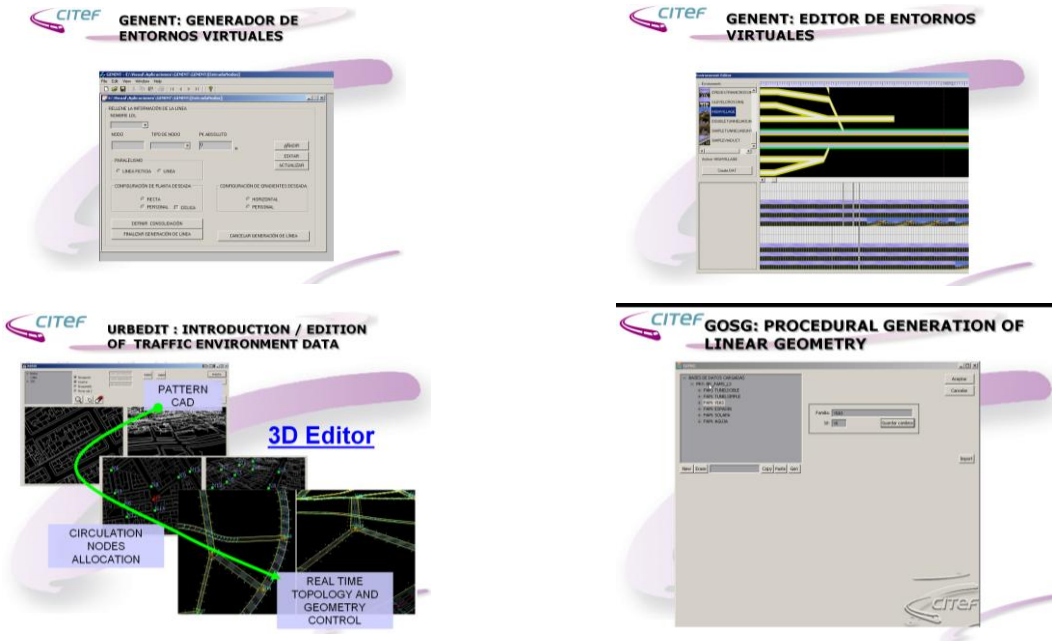


Figura 2.42.Aportaciones de la Tecnología Modular para la creación de los entornos virtuales de los simuladores de CITEF: herramientas para la generación automática de dichos entornos.

3.LA TECNOLOGÍA MODULAR I: MODELADO CONCEPTUAL.

3.1 INTRODUCCIÓN.

Como se comentó en el capítulo anterior, la especificidad de requisitos que acompañan a proyectos de gran envergadura cuya finalidad es una simulación de conducción, obliga al desarrollo de herramientas personalizadas, que satisfagan todas las necesidades con el máximo grado de automatización posible. En este capítulo se describirán se presentará la solución que la Tecnología Modular ha elaborado para la *primera fase de construcción* de un entorno virtual comentada en el Capítulo 1: el Modelado Conceptual.

Se comenzará describiendo las principales técnicas de optimización empleadas en la generación de grandes entornos virtuales y su particularización para el caso que concierne a la presente Tesis, los simuladores de conducción terrestre guiada bajo PC.

A continuación se establecerán las características perceptivas de este tipo de entornos y se analizarán sus repercusiones a la hora de afrontar la problemática asociada a la optimización de la carga gráfica. Los planteamientos llevados a cabo por la Tecnología Modular en este ámbito, sentarán las bases de esta nueva filosofía constructiva del entorno, cuyo pilar constructivo es la trayectoria circulatoria, cuya primitiva de trabajo es el patrón y cuyas herramientas base de optimización son la instanciación y los shaders.

3.2 TÉCNICAS DE OPTIMIZACIÓN EN LA GENERACIÓN DE GRANDES ENTORNOS VIRTUALES.

La generación de grandes entornos virtuales en el mundo de la simulación supone un gran reto. La necesidad de representar una base de datos gráfica de descomunal tamaño con el realismo y las elevadas velocidades de refresco que en este campo se exigen hacen insuficientes a los más innovadores avances del hardware. Es necesario el desarrollo de herramientas de software que ayuden a la disminución de esta potente carga gráfica.

A continuación se exponen los elementos que integran un sistema gráfico y posteriormente se describen las principales técnicas de optimización existentes, prestando especial atención a los niveles de detalle geométricos (*level of detail*, LOD), técnica fundamental en la generación de grandes entornos virtuales y en el desarrollo de la Tecnología Modular.

3.2.1 EL SISTEMA GRÁFICO.

El proceso computacional que comienza con la descripción de los objetos de la escena y termina en la imagen final 2D, se produce mediante una serie de fases consecutivas y conectadas entre si constituyendo el sistema gráfico (*graphics pipeline*). Como indica la Figura 3.1 son tres las etapas en las que se divide este sistema:

- **Etapa de la aplicación**: Se encarga de seleccionar de la base de datos de la escena los objetos que deberán dibujarse en cada fotograma. Esta selección se realiza teniendo en cuenta la visibilidad de cada objeto y definiendo el detalle necesario con el que ha de ser representado. Ambos factores serán descritos más adelante al hablar de las técnicas de recorte de la imagen (visibility culling) y técnicas de adecuación del nivel de detalle (LOD). A continuación los objetos seleccionados serán convertidos en triángulos 3D y finalmente en listas de vértices que serán tratadas con técnicas de compresión para agilizar los cálculos de la GPU.

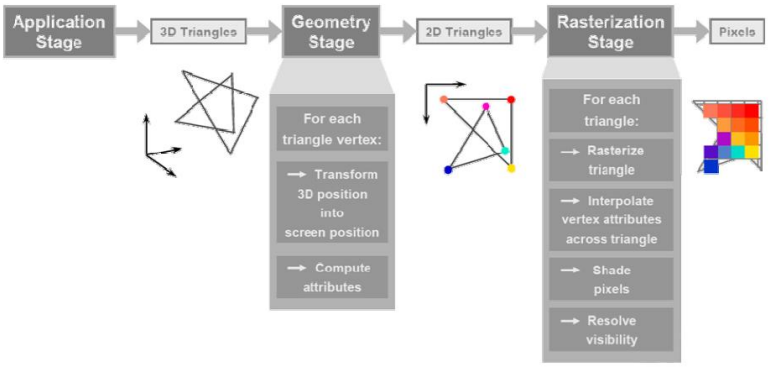


Figura 3.1.El sistema gráfico.

Las Técnicas de compresión [114] generan a partir de un modelo de entrada, un modelo de salida, sustitutivo del de entrada, con el mismo número de símbolos pero codificado de una manera distinta y por tanto sin pérdida de precisión. Existen tanto técnicas de compresión geométrica (Figura 3.2)**,** entre sus aplicaciones destacan el envío de un menor número de vértices a la pipeline gráfica (strips,fans,quads) [115] [116] , como técnicas de comprensión de la imagen.

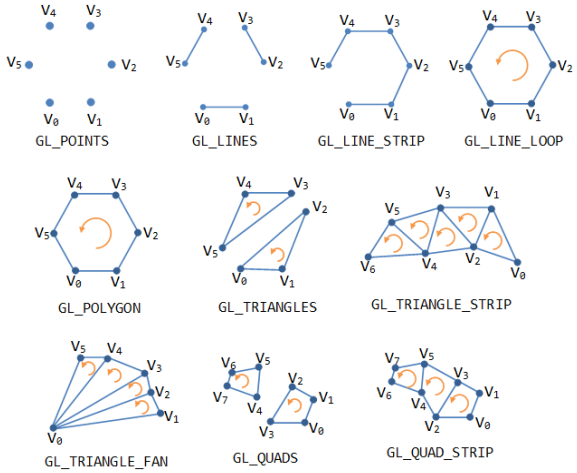


Figura 3.2.Primitivas de OpenGL.

[114] Akenine-Möller, T., Haines,E. Chapter 12. Real-Time Rendering. A.K. Peters Ltd. 2008. ISBN 978-1568814247.
[115] Cai, X. Li,J.,Su, Z. Large-scale terrain rendering using strip masks of terrain blocks. 7th World Congress on Intelligent Control and Automation 2008.

[116] Pouderoux J., Marvie J.-E. Adaptive Streaming and Rendering of Large Terrains using Strip Masks - Proceedings of ACM GRAPHITE 2005, Nouvelle-Zélande.

- **Etapa geométrica**: convierte los triángulos 3D (sus listas de vértices), en triángulos 2D, proyectándolos en la pantalla según el punto de vista. Únicamente las partes de cada primitiva totalmente incluidas en el campo de visión (clipping) pasarán a la siguiente etapa (Figura 3.3).

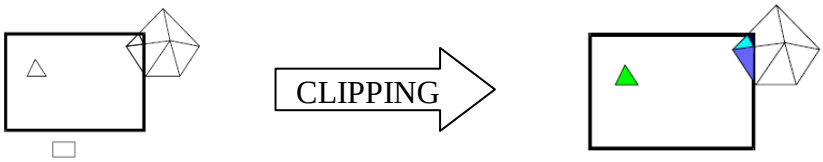


Figura 3.3.Antes y después de la operación de clipping.

- **Etapa de la rasterización**: fragmenta los triángulos 2D en píxeles y calcula el color asociado a cada uno de ellos para constituir así la imagen final. Este color se calcula teniendo en cuenta los atributos que cada vértice tenía asociados en los triángulos 3D iniciales. El algoritmo de z-buffer se encargará de que se rendericen únicamente las zonas visibles.

A medida que el hardware gráfico ha ido evolucionando estas etapas han ido trasladándose de la CPU a la GPU (*Graphics Processor Unit*) (Figura 3.4). Si bien los primeros procesadores gráficos (3dfx Voodoo) solo eran capaces de procesar triángulos 2D, en el 2000 la etapa geométrica se traslada a la GPU (NVIDIA's GeForce 256,GeForce2, ATI's Radeon 7500 y S3's Savage3D) y en el 2001 aparecen los primeros programas sobre la GPU, los shaders. Los shaders [117] permiten la modificación en tiempo real tanto de vértices (*vertex shaders*) como de pixeles (*fragment shaders*), la generación de nueva geometría (*geometry shaders*) y la teselación de la misma en base a un conjunto de bloques (*patches*) definidos por el programador (*tesselation shaders*).

En cuanto al motor gráfico hoy en día existen dos posibles APIs para su desarrollo: DirectX [118] y OpenGL [119]. DirectX, mantenida por Microsoft, solo es válida para Windows y es muy popular en la industria del videojuego. OpenGL, mantenido por el OpenGL Arquitectural Review, o ARB (formado por diversas compañías: 3DLabs, Apple, ATI, Dell, IBM, Nvidia, SGI, Sun), es multiplataforma y es muy popular en el ámbito académico y resto de industrias no dedicadas a los videojuegos. Esta última es la empleada por el motor gráfico desarrollado por CITEF para sus simuladores [120] y el que se empleará en la presente Tesis.

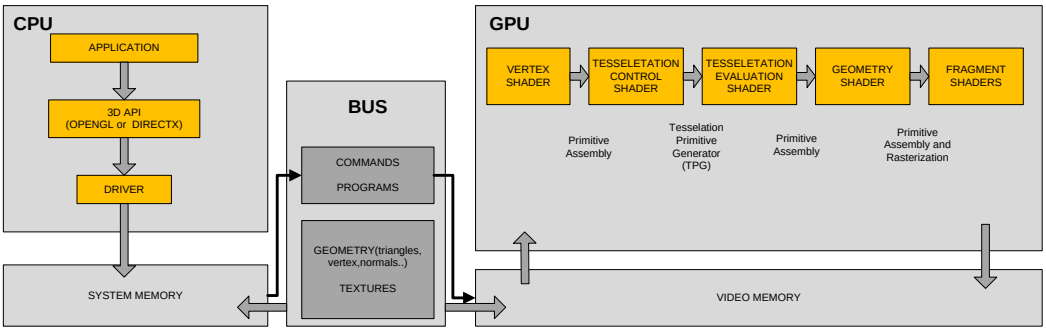


Figura 3.4.Arquitectura del software gráfico.

[117] Wolff, D. OpenGL 4.0 Shading Language Cookbook. Packt Publishing 2011. ISBN: 978-1849514767.

[118] http://windows.microsoft.com/es-ES/windows7/products/features/directx-11 [Última consulta: 9 Enero 2013].

[119] http://www.opengl.org/ [Última consulta: 9 Enero 2013].

[120] Maroto, J. Metodología para la generación de entornos virtuales distribuidos y su aplicación a simuladores de conducción. Tesis (Doctoral), E.T.S.I. Industrial (UPM).Septiembre 2005.

3.2.2 INTRODUCCIÓN A LAS TÉCNICAS DE OPTIMIZACIÓN.

Cada una de las fases que componen el sistema gráfico (*graphics pipeline*) tiene asociado un coste de manera que la frecuencia de salida vendrá determinada por la fase más lenta, produciéndose el efecto de cuello de botella ("*bottleneck*"). Se pueden distinguir tres tipos de coste:

- **Coste de las operaciones por vértice** (proyección, iluminación, clipping): el coste será proporcional al número de vértices que se han enviado desde la base de datos a la escena, el número de luces y el modelo de iluminación. El empleo de diferentes niveles de detalle, permitirá seleccionar en cada momento el número de vértices y propiedades visuales más óptimas.
- **Coste de las operaciones por pixel:** es proporcional al número de pixeles a rellenar y depende también de características del objeto como son el modelo de iluminación, la textura, transparencia, etc.
- **Coste de las transferencias de datos** entre diferentes partes del sistema gráfico o de los accesos a memoria. Este coste es especialmente importante cuando la transferencia de datos se produce a través de un canal compartido por otros recursos del sistema (por ejemplo, el bus de datos), o cuando el sistema debe hacer transferencias de páginas de memoria (por ejemplo, de texturas).

Las técnicas de optimización empleadas hasta el momento para controlar este coste, las podemos englobar en tres categorías:

- **Técnicas de adecuación del nivel de detalle**. Comunmente conocidas como Level of Detail (LOD) [121], se encargan de asignar a cada objeto su nivel de detalle óptimo en la representación virtual. En función de la etapa del sistema gráfico en la que se aplique esta optimización (Figura 3.5) se distingue entre niveles de detalle geométricos, donde se busca una reducción del número de vértices y niveles de detalle no geométricos, muy típicos de los videojuegos, donde se busca una reducción del coste asociado por píxel (*fragment shader*) o por vértice (*vertex shader*).

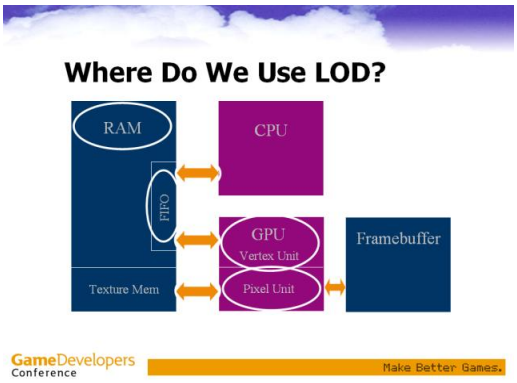


Figura 3.5.Visión esquemática de los posibles empleos de los niveles de detalle (LOD) [122].

- **Técnicas de sustitución de la complejidad geométrica por imagen** (image-based rendering). Con el fin de disminuir el número de polígonos de la escena, se sustituyen

[121] Luebke, R., Varshney, C., Huebner, W. Chapter 6. Level of Detail for 3d Graphics. Morgan Kaufmann Publishers 2003. ISBN 978-1558608382.

[122] Cohen, J., R. Huebner, D. Luebke, M. Reddy, A. Varshney and B. Watson. Level of Detail Management for 3D Games, Tutorial #209, Game Developer Conference 2003, March 4-8, San Jose, CA.

geometrías por imágenes representativas de éstas. Dado que en este caso el sistema de referencia del objeto de entrada y salida no es el mismo, se trata de un sistema de conversión, en lugar de simplificación. Destaca en este ámbito el empleo de impostores (imágenes prerrenderizadas de un objeto complejo que se proyectan sobre un plano que gira con el punto de vista de manera que siempre enfrenten al usuario) [123].

- **Técnicas de recorte de la imagen** (visibility culling)[124]. Eliminan aquéllas partes de la escena que no son visibles desde el actual punto de vista, ya sea porque caen fuera del ángulo de visión (view frustrum culling), porque son ocultadas por otros objetos de la escena (occlussion culling [125]) o porque la normal tiene sentido contrario al punto de vista (backface culling) (Figura 3.6).

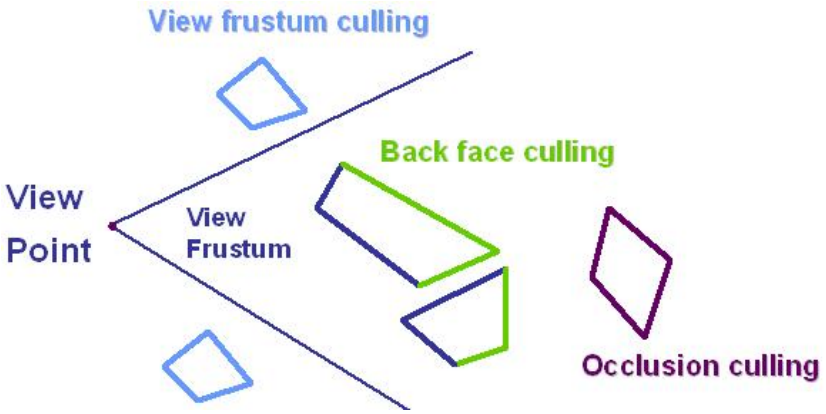


Figura 3.6.Operaciones de culling.

Con el fin de disminuir la carga de trabajo a la hora de procesar la información requerida por la etapa geométrica se emplea un API 3D (OpenGL o DirectX). Ésta consiste en una librería gráfica de más alto nivel que incorpora muchas de las funcionalidades necesarias para llevar a cabo la representación virtual. Algunas de estas librerías gráficas para tiempo real, como OpenSG [126], Open Inventor [127], PLIB [128], SGL [129], OpenRM [130], Open Scene Graph [131], definen estructuras de datos jerárquicas para organizar y optimizar la base de datos con el fin de disminuir el coste de visualización. Son los llamados grafos de la escena (*scene graphs)*. En general todas estas librerías contemplan mecanismos de selección de objetos implementados gracias a funciones automáticas asociadas a los nodos de la estructura jerárquica. Estas estructuras son grafos acíclicos, similares a un árbol pero permitiendo que dos nodos compartan hijos con el fin de

123 Andujar, C., Boo, J., Brunet, P., Fairen, M., Navazo, I., Vazquez, P. and Vinacua A. Omni-directional Relief Impostors. Computer Graphics Forum, Eurographics 2007.Vol. 26, n. 3, pp. 553-560.

[124] Cohen-Or, D., Chrysanthou, Y., Silva C., Durant, F. A Survey of Visibility for Walk-through Applications. IEEE TVCG Journal, July-September 2003. Vol 9, n. 3, pp 412-431.

[125] Mattausch O., Bittner J., Wimmer M. CHC++: Coherent hierarchical culling revisited. Computer Graphics Forum, Proceedings Eurographics 2008. Vol 27, n. 2, pp 221–230.

[126] http://www.opensg.org/ [Última consulta: 9 Enero 2013].

[127] http://oss.sgi.com/projects/inventor/ [Última consulta: 9 Enero 2013].

[128] http://plib.sourceforge.net/ [Última consulta: 9 Enero 2013].

[129] http://sgl.sourceforge.net/ [Última consulta: 9 Enero 2013].

[130] http://www.openrm.org/ [Última consulta: 9 Enero 2013].

[131] http://www.openscenegraph.org/projects/osg [Última consulta: 9 Enero 2013].

ahorrar espacio de almacenamiento. El algoritmo de visualización recorre esta estructura de forma recursiva a partir del nodo raíz y selecciona cuáles son los datos enviados al sistema de procesamiento gráfico.

Estas librerías a su vez controlan la generación de impostores así como las operaciones de culling a partir de la definición de parámetros como son (Figura 3.7):

- Plano de Hither o plano de corte mínimo, considerado como aquel plano perpendicular a la dirección del punto de vista y situado a una distancia d del origen de éste a partir del cual se empezará a representar el escenario.
- Plano de Yon o plano de corte máximo, semejante al anterior salvo que determina el plano a partir del cual se dejará de representar.
- Ángulo de visión: ángulo que forman las esquinas de la pantalla con el origen del punto de vista.

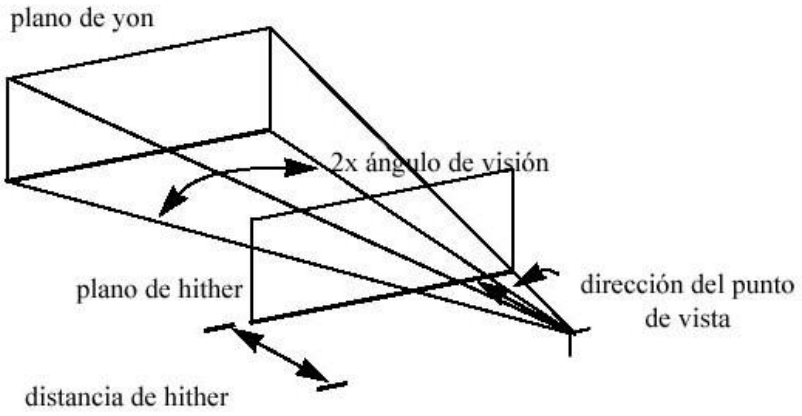


Figura 3.7.Principales parámetros asociados al punto de vista.

El siguiente apartado tratará en mayor profundidad el empleo de niveles de detalle geométricos, por su papel decisivo en la optimización de grandes entornos virtuales y como consecuencia en esta Tesis.

3.2.2.1 NIVELES DE DETALLE GEOMÉTRICOS.

Los componentes de un algoritmo de adecuación del nivel de detalle geométrico son, un modelo multirresolución [132] y una heurística de selección (Figura 3.8). Esta heurística de selección, permite la extracción a partir del modelo multirresolución, del nivel de detalle más adecuado en función de los requerimientos de la aplicación. El nivel de detalle más adecuado es aquel de resolución más simple, que presenta igual apariencia que la representación más detallada. Existen diversos tipos de heurísticas: la distancia al punto de vista, el área ocupada en pantalla, la diferencia de alturas entre el punto medio de la hipotenusa y el asociado al mapa de alturas, llamado error métrico, entre otras. Dichas métricas en ocasiones se combinan.

[132] Danovaro, E., De Floriani, L., Vitali, M., Papaleo, L.: Multi-resolution Morphological Representation of Terrains. Eurographics Italian Chapter Conference 2006, pp 45-51.

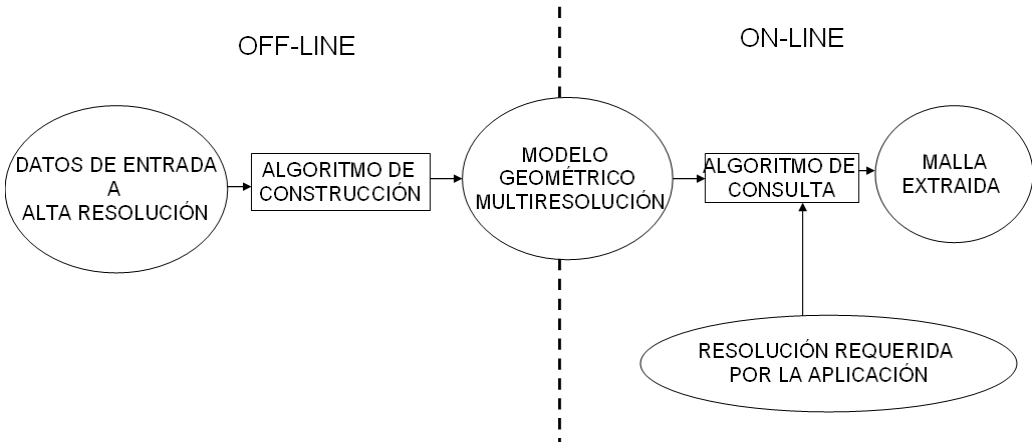


Figura 3.8.Construcción y empleo de una malla multirresolución.

En cuanto al modelo multirresolución, según Andújar [133] se pueden distinguir tres tipos:

- **Colección**: se trata de un conjunto de niveles de detalle independientes y por tanto la memoria ocupada es igual a la suma de memorias de las diferentes representaciones. La extracción de un determinado nivel de detalle es así inmediata.
- **Jerárquico**: mantiene una relación entre los diferentes niveles de detalle consecutivos, de manera que la extracción de un determinado nivel de detalle exige recorrer todos los niveles anteriores.
- **Incremental**: genera una estructura compacta a partir de la cual se pueden extraer diferentes niveles de detalle. Generalmente se almacena la representación de menor detalle y una secuencia ordenada con diferentes actualizaciones. El nivel de detalle deseado se consigue aplicando a la representación más baja el conjunto de actualizaciones necesarias hasta conseguir la precisión deseada.

En función del tipo de algoritmo de LOD que se esté empleando, discreto o continuo [134], se empleará un modelo multirresolución u otro.

Los niveles de detalle discretos (*discrete lod*) se caracterizan por el empleo de modelos multirresolución colección. Se generan independientemente unos de otros en tiempo de precarga, lo que ofrece numerosas ventajas. En primer lugar, esta independencia hace que no sea necesaria la conservación de vértices ni de coordenadas de mapeado entre los diferentes niveles de detalle. Esto posibilita una simplificación más versátil, permitiendo la intervención de un diseñador gráfico. Por otro lado, se minimiza el consumo de recursos en tiempo de ejecución, algo fundamental para aplicaciones que requieren de elevados cálculos en tiempo real, como pueden ser las simulaciones de conducción.

Por el contrario, los niveles de detalle continuos, dependientes o independientes del punto de vista, se fundamentan en el empleo de modelos multirresolución jerárquicos o incrementales. Generalmente el cálculo del modelo se realiza en tiempo de precarga y la extracción del nivel de detalle óptimo en tiempo de ejecución [135]. Este tipo de LOD implica una gran carga de trabajo

[133] Andújar, C.: Geometry simplification. Technical Report LSI-99-2-R, Dept. of LSI, Universidad Politécnica de Cataluña. Feb. 1999.

[134] Luebke, D.P."A Developer's Survey of Polygonal Simplification Algorithms," IEEE Computer Graphics and Applications. May/Jun, 2001.Vol. 21, n. 3, pp. 24-35.

[135] De Floriani, L., Magillo, P. Multiresolution mesh representation: Models and data structures. M. Floater, A. Iske, and E. Qwak, editors. Tutorials on Multiresolution in Geometric Modelling, Springer-Verlag, 2002, pp 363-418.

para la CPU: decidir en tiempo de ejecución qué triángulos se visualizarán en cada instante y con qué detalle y enviarlos a la GPU supone un elevado consumo de recursos.

3.3 APLICACIÓN DE LAS TÉCNICAS DE OPTIMIZACIÓN A LA GENERACIÓN DE GRANDES ENTORNOS VIRTUALES PARA SIMULADORES DE CONDUCCIÓN TERRESTRE BAJO PC.

A la hora de afrontar qué técnicas de optimización emplear en la representación de grandes entornos virtuales resulta fundamental una evaluación previa de las características geométricas y perceptivas de cada elemento. De esta manera suele distinguirse entre técnicas empleadas para la optimización del terreno y técnicas empleadas para la optimización de los restantes elementos situados sobre el mismo.

El terreno, caracterizado por la posibilidad de poseer un amplio campo de visión en el que simultáneamente puedan convivir diferentes niveles de detalle, se caracteriza por permitir el empleo de estructuras de datos muy versátiles, existiendo una gran variedad de algoritmos de adecuación del nivel de detalle (LOD) para su gestión. A lo largo de este apartado se discutirá la elección del algoritmo de LOD más idóneo para el caso tratado en la presente Tesis: los simuladores de conducción terrestre guiada bajo PC.

Por el contrario, a la hora de representar los restantes elementos (mobiliario urbano, elementos de señalización, árboles, etc.) será fundamental determinar la funcionalidad de cada uno de ellos en la escena, lo que estará íntimamente ligado a su nivel perceptivo visual. El empleo de texturas que simplifiquen la complejidad geométrica por imagen, unido a una adecuada estructuración en el grafo de la escena será fundamental en estos casos.

3.3.1 OPTIMIZACIÓN DEL TERRENO.

La principal herramienta para combatir la elevada carga gráfica del terreno será el empleo de algoritmos de adecuación del nivel de detalle. Será necesario optar por el empleo de un nivel de detalle continuo o discreto. El papel prioritario que juegan las trayectorias en las simulaciones de conducción terrestre es fundamental en esta decisión. El alto grado de detalle con el que es necesario visualizarlas condiciona las estructuras de datos a emplear e incrementa los cálculos necesarios.

3.3.1.1 NIVEL DE DETALLE CONTINUO.

En el caso de los niveles de detalle continuos, los modelos multirresolución pueden dividirse en dos grupos atendiendo a la estructura de datos que empleen: regulares e irregulares [136]. Ambas estructuras darán lugar a diversos modelos multirresolución que diferirán en la forma de las celdas, la arquitectura para su descomposición espacial y la jerarquía inducida por ésta. Todas estas características hacen que un determinado modelo multirresolución sea más o menos adecuado para una determinada aplicación, ya que no existe un modelo multirresolución que satisfaga todas las necesidades.

[136] De Floriani, L., Magillo, P. Regular and Irregular Multiresolution Terrain Models: a Comparison. Proceedings of the 10th ACM international symposium on Advances in geographic information systems, November 08-09, 2002, McLean, Virginia, USA.

MODELOS REGULARES:

Se fundamentan en una descomposición jerárquica regular. La mayor compacidad de los modelos jerárquicos los hace más eficientes a la hora de extraer el nivel de detalle deseado y a la hora de implementar diferentes funciones necesarias en una aplicación de realidad virtual: tests de visibilidad, determinación de colisiones, deformaciones dinámicas..etc. Ejemplos típicos de modelos multirresolución son el *quadtree* [137] [138] (Figura 3.9), el árbol binario de triángulos (*binary triangle tree, bintree*) [139] (Figura 3.10) y el *octree* [140].

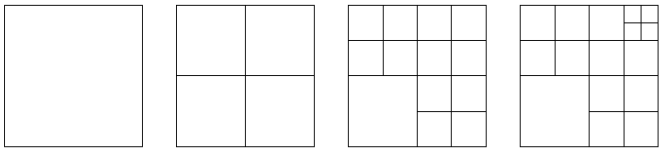


Figura 3.9.Quadtrees

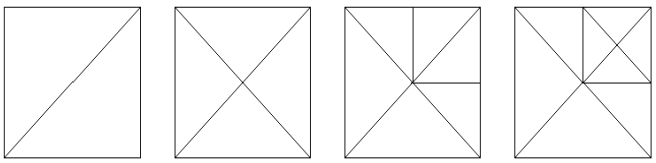


Figura 3.10.Binary triangle trees (bintree).

En este ámbito destacan los algoritmos atribuidos a Lindstrom et al. 1996 [141], Duchaineau et al. 1997 (ROAM) [142] , Röttger et al. 1998 [143] y Lindstrom & Pascucci 2001 [144] .

La siguiente figura muestra un ejemplo del algoritmo de ROAM, uno de los más populares para la representación de terrenos. Emplea como estructura jerárquica un árbol binario de triángulos (bintree) y modifica la resolución de cualquier parte de la malla empleando operaciones división (*split*) y fusión (*merge*). La división de cada triángulo se realiza mediante la bisección de éste por su lado más largo, añadiendo un nuevo vértice en el punto medio de la hipotenusa, dando así lugar a dos nuevos triángulos isósceles.

---

[137] Leclerc, Y. G. and Lau, S. Q. TerraVision: A Terrain Visualization system. Technical Note 540, SRI International 1994, pp 1–20.

[138] Rottger, S., Heidrich, W., Slussallek, P., Seidel, H-P.: Real-Time Generation of Continuous Levels of Detail for Height Fields, Proc. 6th Int. Conf. in Central Europe on Computer Graphics and Visualization, 1998, pp. 315-322.

[139] Lindstrom, P., Koller, D., Ribarsky, W., Hodges, L.F, Faust, N. and Turner, G.A. Realtime, continuous level of detail rendering of height fields. In Proceedings SIGGRAPH 96. pp 109-118.

[140] Fairen, M.; Trueba, R. Octree-based view-dependent triangle meshes. In Proceedings WSCG 2007. 15th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, pp. 193-200.

[141] Lindstrom, P, Koller, D., Ribarsky, W., Hodges, L.F, Faust, N. and Turner, G.A. Realtime continuous level of detail rendering of height fields. In Proceedings SIGGRAPH 96, pp 109-118.

[142] Duchaineau, M., Wolinsky,M, Sigeti, D.E., Miller, M.C., Aldrich, C. and Mineed-Weinstein, M.B. Roaming terrain: Real-time optimally adapting meshes. In Proceedings IEEE Visualization 1997, pp 81-88.

[143] Rottger, S., Heidrich, W., Slussallek, P., Seidel, H-P. Real-Time Generation of Continuous Levels of Detail for Height Fields. In Proceedings of 6th Int. Conf. in Central Europe on Computer Graphics and Visualization, 1998, pp. 315-322 .

[144] Lindstrom, P. and Pascucci, V. Visualization of large terrains made easy. In Proceedings of IEEE Visualization 2001, pp 363-370.

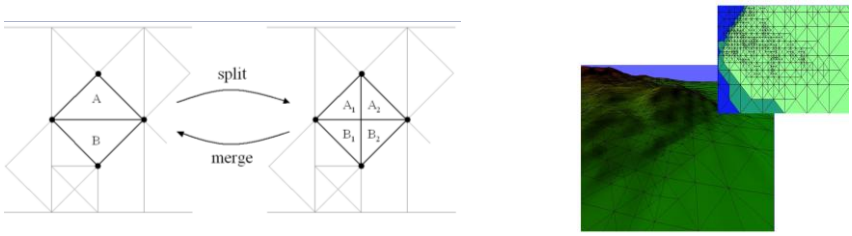


Figura 3.11.Ejemplo del algoritmo de ROAM: operaciones de división (split) y fusión (merge).

Posteriormente se han realizado mejoras de estos algoritmos, como la propuesta por Strugar 2009 [145], que resuelve algunos de los problemas asociados a los quatrees, como los relacionados con las discontinuidades en las transiciones entre niveles de detalle. Wang et al 2011 [146] proponen un algoritmo basado en quatrees que mejora el tratamiento de elementos sobre el terreno.

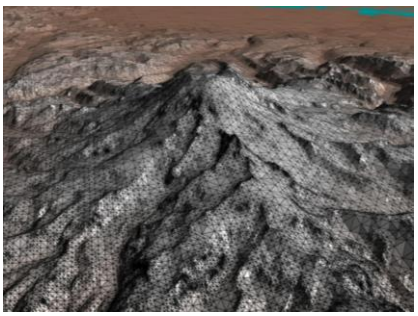


Figura 3.12.Representación de Puget Sound, Washington ofrecida por Lindstrom & Pascucci 2001.

En cualquier caso se trata de algoritmos computacionalmente costosos orientados a representaciones que no estén limitadas en este sentido. Se han realizados diversos intentos para su aplicación en representaciones en tiempo real, como la propuesta por Bryan Turner [147], implementada en el motor gráfico TreadMarks [148] y la propuesta por Mathew White 2008 [149], que realizan una modificación del algoritmo de ROAM para su empleo en videojuegos.

MODELOS IRREGULARES:

Estos modelos generan aproximaciones simplificadas de la superficie del terreno empleando para ello mallas irregulares de triángulos (*Triangular Irregular Network*, TIN). Desde el punto de vista de la representación gráfica las TINs presentan la ventaja de, para un determinado grado de precisión deseado, aproximar el modelo con un menor número de triángulos. sin embargo, sus estructuras de almacenamiento más complejas dificultan la obtención de un modelo multirresolución y las consultas en tiempo real. Por otro lado, la TIN es

[145] Strugar, F. Continuous Distance-Dependent Level of Detail for Rendering Heightmaps. In Proceedings of J. Graphics, GPU, & Game Tools 2009, pp 57-74.

[146] Wang, L., Zhao, S. Lu, Y. Multi-resolution quad-tree based algorithm for real- time visualization of massive terrain dataset. International Conference on Multimedia Technology (ICMT) 2011, pp 5934 - 5938

[147] Turner, B. Real-Time Dynamic Level of Detail Terrain Rendering with ROAM. Available at: http://www.gamasutra.com/view/feature/3188/realtime_dynamic_level_of_detail_.php [Última consulta: 9 Enero 2013].

[148] http://www.TreadMarks.com. [Última consulta: 9 Enero 2013].

[149] White, M. Real-Time Optimally Adapting Meshes: Terrain Visualization in Games. International Journal of Computer Games Technology 2008. Vol. 2008.

la estructura más eficiente para la inserción de elementos lineales, de ahí su importancia en la creación de entornos virtuales para simuladores de conducción terrestre.

Los primeros modelos desarrollados en este ámbito se basaban en jerarquías de mallas irregulares [150] o en grupos de mallas, cada una con una resolución uniforme diferente, que se conectaban a través de una serie de interfaces de unión [151] [152]. Posteriormente aparecieron los modelos progresivos que codificaban una malla a baja resolución junto a una serie de modificadores que permitieran ir añadiendo más precisión [153] [154]. En este ámbito destacan los algoritmos atribuidos a Hoppe 98 [155] (Figura 3.13) y DeFloriani et al. 2000 [156].

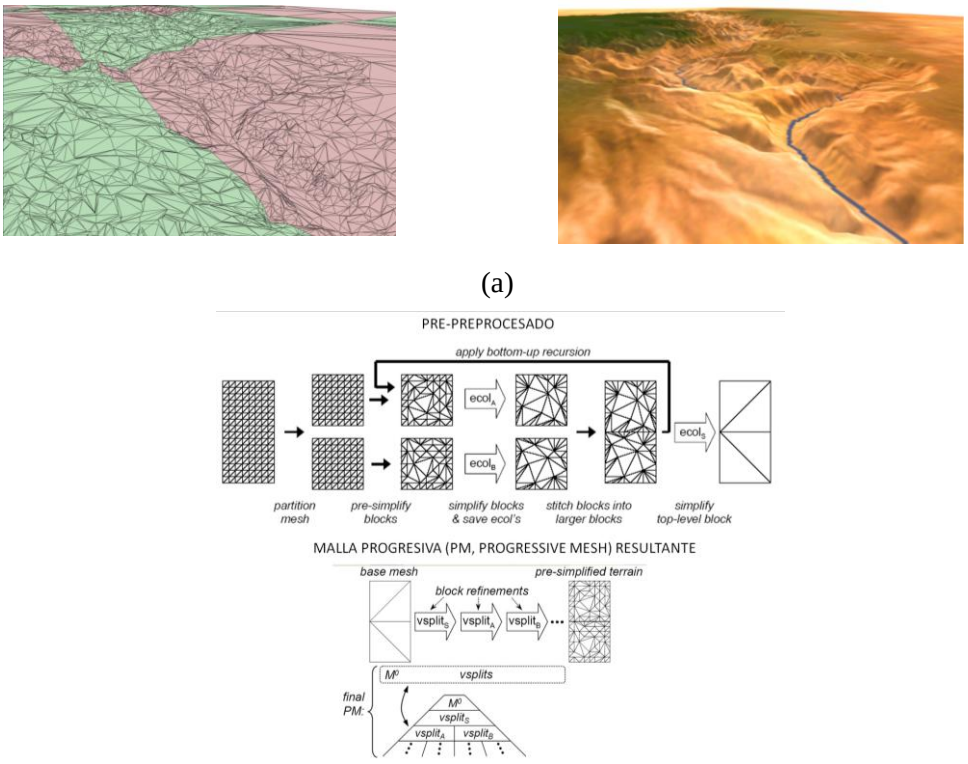


Figura 3.13.(a) Representación del Gran Cañon. (b) Proceso constructivo de la malla progresiva de Hoppe 98: codificación de la malla a partir de un modelo simplificado (M0) y un conjunto de modificadores (vsplits).

[150] De Floriani, L., Puppo, E. Hierarchical triangulation for multiresolution surface description. ACM Transactions on Computers, 1995.Vol 14, n. 4, pp 363-411.

[151] De Berg, M., Dobrindt, K.. On levels of detail in terrains. In Proceedings 11th ACM Symposium on Computational Geometry, Vancouver (Canada), ACM Press, 1995, pp C26-C27 .

[152] De Floriani, L. A pyramidal data structure for triangle-based surface description. IEEE Computer Graphics and Applications, 1989.Vol 8, n.2 , pp 67-78.

[153] Hoppe, H. Progressive meshes. In Proceedings of the ACM SIGGRAPH 1996. pp 99-108.

[154] Taubin, G. , Gueziec, A., Horn, W., Lazarus, F. Progressive forest split compression. In Proceedings of the ACM SIGGRAPH, 1998, pp 123-132.

[155] Hoppe, H. Smooth view-dependent level-of-detail control and its application to terrain rendering. In Proceedings IEEE Visualization 1998, pp 35–42.

[156] De Floriani, L. ,Magillo,P., Puppo, E.: VARIANT: A System for Terrain Modeling at Variable Resolution. GeoInformatica 2000. Vol. 4, n. 3, pp 287-315.

Intentando extraer lo mejor de los modelos regulares e irregualres se han desarrollado modelos híbridos [157] [158] donde el terreno, definido a partir de una distribución irregular de puntos, es estructurado en un quadtree (QuadTIN). En cada paso de subdivisión del quadtree, la diagonal de cada cuadrilátero no tiene porque ser dividida necesariamente por su punto medio, sino que se emplea el punto más próximo a éste dado en la distribución irregular de puntos.

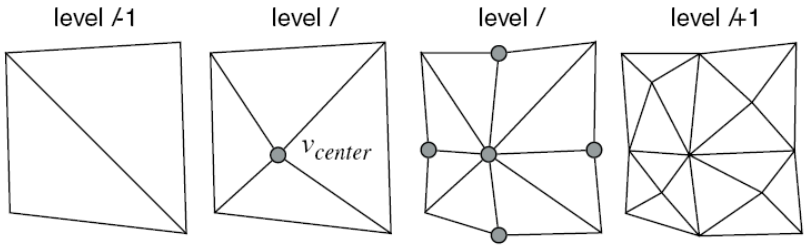


Figura 3.14.Ejemplo de subdivisión en una QuadTIN.

La falta de restricciones de estos modelos irregulares los hace computacionalmente costosos, debido a que tienen que gestionar las relaciones entre los triángulos de la malla y mantener sus dependencias durante el refinamiento o la simplificación de la misma. Además, como la posición de los vértices no viene implícita en la definición del modelo, como ocurre en los modelos regulares, éstos deben almacenar todas las coordenadas de los vértices de las mallas y no sólo el valor de la altura, ocupando más memoria.

Por otro lado, este tipo de modelos no está pensado para la inserción de elementos lineales, ya que exigiría el empleo de una triangulación de Delaunay restringida [159] que incrementalmente se fuese actualizando en función de la variación del punto de vista [160], lo que incrementaría mucho los costes computacionales. En su lugar, los elementos lineales son introducidos generalmente mediante texturas, ofreciendo un nivel de detalle insuficiente para un simulador de conducción terrestre.

3.3.1.2 NIVELES DE DETALLE DISCRETOS.

Los niveles de detalle discretos reúnen una serie de características que los convierten en una alternativa interesante a tener en cuenta a la hora de representar el terreno. Entre estas características destacan:

- una simplificación más versátil como consecuencia de la independencia entre niveles de detalle:
    - no es necesario la conservación de vértices entre los diferentes niveles.
    - posible intervención de un diseñador gráfico.

[157] Feng,Z., Qi, H. .A Hybrid LOD Algorithm Based on TIN Modelling. XXIst ISPRS Congress Technical Commission II 2008. Vol 37, n. 2, pp 953-956

[158] Lario, R., Pajarola, R., Tirado, F.: Hyperblock-QuadTIN: Hyperblock quadtree based triangulated irregular networks. In Proceedings IASTED International Conference on Visualization, Imaging and Image Processing (VIIP) 2003, pp. 733–738.

[159] Chew, L. P. Constrained Delaunay triangulations. In Proceedings of the Third Annual Symposium on Computational Geometry (Waterloo, Ontario, Canada, June 08 - 10, 1987). D. Soule, Ed. SCG '87. ACM Press, New York, NY, pp 215-222.

[160] R. Klein , D. Cohen-Or , T. Huttner. Incremental view-dependent multiresolution triangulation of terrain. Proceedings of the 5th Pacific Conference on Computer Graphics and Applications, October 13-16, 1997.

  - óptima introducción de elementos lineales (carreteras, vías ferroviarias, etc.).

- Una minimización del consumo de recursos de CPU debido a la no evaluación a nivel de vértice de la métrica de selección.

Estos dos últimos puntos resultan cruciales al hacer frente una representación virtual en una simulación de conducción terrestre. A su vez, presentan una serie de inconvenientes:

- Necesidad de fragmentación del terreno en bloques (tiles), lo que supone por un lado la introducción de polígonos innecesarios y por otro lado la posible percepción de los cambios de nivel de detalle (agujeros y saltos bruscos entre los diferentes niveles de detalle, conocidos como popping).
- No optimización a nivel de vértice en función del punto de vista (en la presente Tesis se aportará una solución de compromiso a este problema).

En el ámbito de los niveles de detalle discretos una solución a destacar es la propuesta en Smartmesh [161]. Smartmesh fue desarrollado por Terrex (hoy en día parte integrante de Presagis, bajo el kit de herramienta de Terra Vista [162]), una de las aplicaciones comerciales para la generación de terreno más difundida a nivel profesional. Aprovechan todas las ventajas de los niveles de detalle discretos y evitan los cambios bruscos de nivel de detalle mediante una triangulación especial en los bordes de sus bloques. Ésta que será comentada en el Capítulo 5 al tratar el posicionamiento modular.

3.3.1.3 NIVELES DE DETALLE CONTINUOS VERSUS NIVELES DE DETALLE DISCRETOS EN SIMULACIONES DE CONDUCCIÓN TERRESTRE.

Las ventajas que ofrecen los niveles de detalle continuos que operan a nivel de vértice son evidentes:

- simplificación a nivel de vértice y no de objeto y por tanto un nivel de detalle más optimizado en función del punto de vista.
- posibilidad de evitar los saltos bruscos entre diferentes niveles de detalle (popping) mediante la introducción de funciones de suavizado (geomorphs).

Sin embargo, a la hora de afrontar la generación de terrenos para la simulación de conducción terrestre con toda la red de trayectorias que esto supone, se han de hacer frente graves inconvenientes:

- Implementación compleja.
- Dificultosa optimización del entorno: complicada introducción de elementos lineales precisos, no están pensados para este fin (aumentan considerablemente el número de cálculos necesarios, suponiendo un mayor consumo todavía de CPU). Sin embargo, la proximidad del conductor a su entorno en este tipo de aplicaciones hacen que las trayectorias y sus alrededores adquieran un papel primordial.
- Implican una elevada sobrecarga de memoria y CPU en tiempo de ejecución:
  - estos algoritmos pueden soportar la instanciación, pero de una manera muy ineficiente, ya que cada instancia debe almacenar su lista de vértices activos. Sin

[161] Willis, L. Who says you can't teach an old LOD new tricks. In Proceedings Image 1998, The Image Society, Chandler, Arizona.

[162] http://www.presagis.com/products_services/products/modeling-imulation/content_creation/terra_vista/ [Última consulta: 9 Enero 2013]

embargo, en los entornos de conducción terrestre muy a menudo se encuentran elementos repetitivos que podrían instanciarse.

- en lugar de seleccionarse un nivel de detalle por cada objeto visible, los algoritmos de LOD continuos dependientes del punto de vista han de evaluar cada vértice activo en cada objeto visible y actuar acorde a la métrica de selección (simplificando regiones de objetos que caen fuera del campo de visión, en lugar de ignorarlos). Sin embargo, en este tipo de aplicaciones virtuales el conocimiento a priori de las trayectorias de circulación permiten conocer en tiempo de precarga las zonas que será necesario optimizar con mayor y menor detalle.

- Finalmente, no explotan los recursos disponibles por las actuales GPUs. Estos modelos multirresolución no fueron diseñados para sacar el máximo partido al actual hardware. La mayor velocidad a la que evolucionan las GPUs frente a las CPUs es la responsable de esta desactualización. El cuello de botella al aplicar estos algoritmos ha dejado de ser el número de polígonos y ha pasado a ser el bus de comunicación CPU-GPU. Éste no es capaz de enviar a la GPU toda la información que es capaz de representar. No hay que olvidar que la representación del terreno no es más que una de las múltiples tareas que la CPU tiene que gestionar en un simulador de conducción.

Con el fin de disminuir la carga de CPU y maximizar los recursos ofrecidos por las actuales GPUS, se comenzaron a desarrollar un nuevo tipo de algoritmos cuya primitiva de trabajo deja de ser el triángulo [163] [164] [165] [166] [167]. En su lugar se emplean bloques de triángulos, optimizados en tiempo de precarga de manera que se garantice una óptima organización espacial de éstos (*stripping*) agilizando así los cálculos de la GPU

Las mejoras en este nuevo tipo de algoritmos no dejan de sucederse [168] [169] [170] [171], sin embargo, las triangulaciones que generan estos algoritmos necesitan mucho espacio de almacenamiento e implican frecuentes accesos a disco que pueden disminuir la velocidad de refresco. Por otro lado, estos algoritmos infrautilizan las singularidades que caracterizan a los simuladores de conducción terrestre bajo PC y que pueden contribuir a la optimización de estos entornos. Estas singularidades han sido estudiadas por esta Tesis y empleadas como base para desarrollar una nueva metodología constructiva de entornos virtuales, la Tecnología Modular, que gracias a su primitiva de trabajo, el módulo y a su sistema de instanciación y empleo de shaders

[163] Pomeranz, A. Roam using surface triangle clusters (RUSTIC). Master's thesis, University of California at Davis, 2000.

[164] Levenberg, J. Fast view-dependent level-of-detail rendering using cached geometry. In Proceedings IEEE Visualization '02, pp 259–266.

[165] Lario, R., Pajarola, R., Tirado, F. Hyperblock-QuadTIN: Hyper-block quadtree based triangulated irregular networks. In IASTED International Conference on Visualization, Imaging and Image Processing, VIIP 2003.

[166] Schneider, J. Westermann,R. GPU-Friendly High-Quality Terrain Rendering. Journal of WSCG, Plzen, Czech Republic, 2006.

[167] Gobbetti, E. et al. C-BDAM - Compressed Batched Dynamic Adaptive Meshes for Terrain Rendering. In Proceedings Computer Graphics Forum, 2006. Vol. 25, n. 3, pp. 333-342.

[168] Vanek,J.,Jezek, B. Practical Algorithm for Unlimited Scale Terrain Rendering.2nd European Conference of Computer Science (ECCS '11).Tenerife, Spain. 2011.

[169] Livny, Y., Kogan, Z. and El-Sana, J. Seamless patches for GPU-based terrain rendering. In Proceedings of The Visual Computer 2009, pp 197-208.

[170] Dick, C., Schneider, J. and Westermann, R. Efficient Geometry Compression for GPU-based Decoding in Realtime Terrain Rendering. In Proceedings of Comput. Graph. Forum 2009, pp 67-83.

[171] Gobbetti,E., Marton,F. Cignoni,P., Di Benedetto,M. and Ganovelli, F. C-BDAM - Compressed Batched Dynamic Adaptive Meshes for Terrain Rendering. In Proceedings Computer Graphics Forum, 2006. Vol .25, n.3, pp 333-342.

sacan partido a las capacidades de las actuales GPUs a la vez que disminuyen los espacios de almacenamiento requeridos por los algoritmos vistos.

3.3.2 OPTIMIZACIÓN DE LOS RESTANTES ELEMENTOS DEL ESCENARIO.

Como se comentó anteriormente, las diferentes condiciones que reúnen los elementos que han de posicionarse sobre el terreno (mobiliario urbano, edificios, arboles, señalización..etc) frente al terreno, tanto desde un punto de vista perceptual como de carga geométrica, exigen un tratamiento diferenciado [172].

El elevado consumo de CPU que supone el empleo de niveles de detalle continuos en su optimización descarta el empleo de los mismos. Su optimización pasa por su estructuración jerárquica en un grafo de la escena y el empleo de niveles de detalle discretos, siendo decisiva la determinación del número óptimo de nodos del árbol. Un recorrido de éste demasiado complejo puede provocar el efecto contrario, una ralentización de la velocidad de refresco.

Este diseño óptimo del grafo de la escena implicará tomar decisiones sobre el número de niveles de detalle discretos necesarios así como el número de instancias. A su vez, para cada nivel de detalle será necesario definir, la carga poligonal y las distancias de corte más adecuadas, con el fin de evitar transiciones bruscas (*popping*) que puedan distraer al conductor.

Dentro de los entornos abiertos, merecen especial atención los entornos urbanos por sus particulares características [173] [174] [175]. A pesar de tratarse de entornos abiertos, los edificios pueden provocar una gran reducción del campo de visión, viéndose así muy beneficiados por la aplicación de técnicas de *culling* que disminuyan drásticamente la carga gráfica [176] [177].

Son muy numerosas las investigaciones llevadas a cabo en torno a la automatización en la construcción y optimización geométrica de los edificios [178] [179], estando enfocado su empleo a la representación virtual de ciudades empleadas con fines inmobiliarios, turísticos o publicitarios cuya velocidad de refresco dista mucho de ser la exigida en una simulación de conducción. En este último caso, el escaso valor funcional de estos elementos de cara a la conducción y por tanto, la baja percepción que el usuario puede tener de ellos, no justifica el consumo de recursos que implica su empleo siendo más adecuada la sustitución de la geometría por imagen, como puede

[172] Yang, L., Zhang, L., Ma, J., Xie, J. and Liu, L. Interactive visualization of multi-resolution urban building models considering spatial cognition. In Proceedings of International Journal of Geographical Information Science 2011. Vol 25, n. 1, pp 5-24.

[173] Benedetto, M.D., Cignoni, P., Ganovelli, F., Gobbetti, E., Marton, F. and Scopigno, R. Interactive Remote Exploration of Massive Cityscapes. In Proceedings of VAST. 2009, pp 9-16.

[174] Day, A.M.; Arnold, D.B.; Havemann, S.; Fellner, D. Combining polygonal and subdivision surface approaches to modeling urban environments. Computers & Graphics 2004. Vol 28, n. 4, pp 497–50.

[175] Royan, J.C.Bouville, P.Gioia. PBTree- A new porgessive and hierarchical representation for network based navigation in urban environments. Vision Modeling and Visualization. Munich, Germany, 2003, pp. 299–307.

[176] Downs, L., Möller T., Séquin, C.H. Occlusion horizons for driving through urban scenery, Proceedings of the 2001 symposium on Interactive 3D graphics, pp.121-124,

[177] Andújar, C., Saona-Vázquez, C., Navazo, I., Brunet, P. Integrating occlusion culling with levels of detail through hardly-visible sets. In Comp. Graphics Forum 2000. Vol 19, n. 3, pp 499-506.

[178] Döllner, J, Buchholz, H. Continuous level-of-detail modeling of buildings in 3D city models. In Proceedings of ACM GIS 2005, pp. 173-181.

[179] Gobbetti, E., Marton, F., Benedetto, M.D., Ganovelli, F., Bühler, M., Schubiger, S., Specht, M., Engels, C. and Gool, L.J.V. Reconstructing and Exploring Massive Detailed Cityscapes. In Proceedings of VAST. 2011, pp 1-8.

ser mediante el uso de impostores [180]. El empleo de texturas se convierte así en una técnica indispensable que permitirá suplir los detalles geométricos de todos aquellos elementos que ocupen un papel secundario en la representación virtual, como pueden ser fachadas de edificios y árboles.

La composición de texturas es otra técnica muy empleada (Figura 3.15). Consiste en la introducción en una única textura toda la información de texturas correspondiente a un determinado modelo o incluso de varios modelos. De esta manera se disminuyen el número de cambios de estado necesarios en el renderizado.

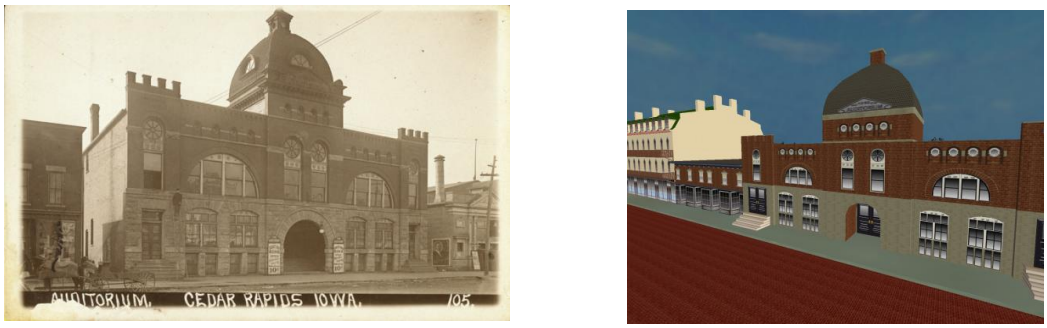


Figura 3.15.Ejemplo de sustitución de detalles geométricos por imagen.

3.4 PRINCIPIOS DE LA TECNOLOGÍA MODULAR.

Desde que en 1976 James H. Clark sentara las bases de las principales técnicas de optimización [181], son innumerables los avances conseguidos y los matices que cada técnica tiene en su implementación en función del tipo de aplicación a la que se destine y del hardware disponible. Por otro lado, los estudios en percepción visual tienen una gran relevancia a la hora de optimizar una base de datos gráfica [182] [183] [184] [185]. Son factores muy importantes a tener en cuenta:

- La excentricidad: el ojo humano solo es capaz de percibir con detalle los objetos que se encuentran en una pequeña franja del campo de visión (2 grados en la región foveal) por lo que resulta innecesario representar con gran precisión todo aquello que caiga fuera de este ángulo.
- La velocidad: el ojo humano percibe con menor detalle los objetos en movimiento.

[180] Andujar C.,Díaz J. and Brunet P. Relief Impostor Selection for Large Scale Urban Rendering. IEEE Virtual Reality Workshop on Virtual Citiscapes: Key Research Issues in Modeling Large-Scale Immersive Urban Environments. 2008.

[181] Clark, J. H. Hierarchical geometric models for visible surface algorithms. Communications of the ACM, 1976. Vol. 19, n.10, pp 547-554.

[182] Hasic, J., Chalmers, A., Sikudova, E.: Perceptually guided high-fidelity rendering exploiting movement bias in visual attention. ACM Transactions on Applied Perception 2010. Vol. 8, n. 1, pp 1-19.

[183] Bartz, D.; Cunningham, D.W.; Fischer, J.; Wallraven, C. The Role of Perception for Computer Graphics.In Proceedings of the 29th Annual Conference Eurographics 2008, pp 65-86.

[184] Cater, K., Chalmers, A., and Ward, G. Detail to Attention: Exploiting Visual Tasks for Selective Rendering. In Proceedings of the Eurographics Symposium on Rendering, 2003, pp 270–280.

[185] Yee, H., Pattanaik, S., Greenberg, D. Spatiotemporal sensitivity and Visual Attention for efficient rendering of dynamic Environments. In ACM Transactions on Computer Graphics 2001. Vol. 20, n. 1, pp. 39–65.

- Existencia de rasgos sobresalientes: investigaciones en el campo de la psicología [186] [187] [188] han puesto de manifiesto que el sistema visual humano es altamente sensible ante determinadas características tales como, cambios bruscos de color, movimientos repentinos, los bordes de los objetos. Será por tanto necesario tratarlos con especial interés de manera que no causen distracciones no planeadas.
- La atención: La atención visual es el resultado de una acción coordinada que tiene lugar en el cerebro, donde se ven involucrados tanto procesos voluntarios como involuntarios y que nos permite centrarnos de una manera rápida y eficaz en la información más relevante. Cuando se requiere información detallada de diversas áreas de un determinado entorno, el ojo humano no escanea la escena de una manera raster, sino que va saltando de manera que los objetos más destacados van cayendo en el interior de la región foveal. El cerebro posteriormente ensambla toda esta secuencia y genera una visión del entorno coherente aunque imperfecta. Según experimentos llevados a cabo por Cater et al [189], cuando el usuario de una determinada aplicación virtual centra su atención en una tarea específica, el resto del universo pasa para él desapercibido. Es la llamada "ceguera por falta de atención" ("*Inattentional Blindness*"). De esta manera, el conocimiento a priori de estos objetos protagonistas de la escena (task objects), puede contribuir a una gran simplificación en la representación del entorno, sin que el usuario llegue a percibirlo.

Al abordar la optimización de un gran entorno virtual dedicado a la simulación de conducción terrestre bajo PC, se ha de afrontar en primer lugar una base de datos gráfica de descomunal tamaño y a su vez, una serie de dificultades específicas de este tipo de representaciones virtuales:

- **Gran proximidad del conductor a su entorno**, lo que implica una necesidad de gran realismo en las proximidades de las trayectorias. Esto únicamente puede conseguirse mediante la inserción geométrica de las trayectorias en el terreno, lo que tiene fuertes repercusiones en la selección del modelo de optimización a emplear. La inserción de dicha geometría implica un reajuste y retriangulación de la malla del terreno en los alrededores, con el fin de adaptarse a la mayor resolución de la trayectoria.
- **Necesidad de elevadas velocidades de refresco**, que permitan circular a través del entorno a la velocidad deseada por el conductor sin producirse tirones: el ojo humano es capaz de percibir un máximo de unos 30 fotogramas/s. Por debajo de este umbral se pierde la sensación de realismo y superar este umbral supone un derroche de recursos, que puede destinarse a otros fines, como la simulación de la dinámica vehicular y las comunicaciones. El hardware disponible establecerá el límite en el número de polígonos y texturas a renderizar y a su vez determinará la implementación más idónea de dichas técnicas, ya que a medida que las GPUs evolucionan lo hacen su modo y capacidad de almacenamiento y procesamiento.
- **Elevado consumo de CPU por parte del módulo de conducción**, lo que exige el empleo de técnicas de optimización ahorrativas en este sentido. Nuevamente los niveles de detalle discretos se adecuan más a estas restricciones que los niveles de detalle continuos.

[186] Itti, L. and Koch, C.. A saliency-based search mechanism for overt and covert shifts of visual attention. In Vision Research, 2000, Vol. 40, n. 10-12.

[187] Yantis, S. Attentional capture in vision. In A. Kramer, M. Coles, & G. Logan (Eds.), Converging operations in the study of selective visual attention. Washington, DC: American Psychological Association 1996, pp. 45-76.

[188] Yarbus, A.L. Eye movements during perception of complex objects. In Eye Movements and Vision, 1967, Plenum Press, New York, Chapter VII, pp. 171–196.

[189] Cater, K., Chalmers, A., Ledda, P. Selective quality rendering by exploiting human inattentional blindness: Looking but not seeing. In Symposium on Virtual Reality Software and Technology 2002, pp 17-24.

- **Minimización de los tiempos de carga**, en las simulaciones de conducción terrestre destinadas al entrenamiento de conductores es necesario poder lanzar diversos ejercicios en diversos entornos de ejecución de una manera cómoda y eficaz, lo que implica que los tiempos de carga no puedan ser muy elevados.

A su vez, este tipo de simuladores presentan una serie de características que pueden favorecer en su optimización y a las que no suele sacarse todo el provecho que debiera:

- **El conocimiento a priori de los elementos del entorno** que van a jugar un papel fundamental en la representación virtual (task objects) va a permitir determinar que elementos será necesario modelar con un mayor detalle: trayectorias, señalización y mobiliario que afecte directamente a la conducción, como estaciones o paradas.
- **El conocimiento de las trayectorias de circulación** permite establecer la distancia a la que serán observados los distintos elementos, definiendo así en tiempo de precarga los parámetros de los niveles de detalle más óptimos: distancias de corte y carga poligonal de cada nivel de detalle. Se ve así favorecido el empleo de niveles de detalle discretos.
- **Las elevadas velocidades de circulación** a través del entorno así como la prioridad del usuario por realizar una conducción correcta, dificultan la percepción de éste 190 191.
- **La repetibilidad** de numerosos elementos del entorno, unido a la baja percepción que el usuario tiene de ellos, permite explotar las ventajas de la instanciación. Nuevamente se ve así favorecido el empleo de niveles de detalle discretos, ya que los niveles de detalle continuos no son capaces de manejar con efectividad la instanciación.

La presente Tesis ha materializado todo este análisis en una nueva metodología para la construcción de entornos virtuales, la denominada Tecnología Modular. Dicha tecnología genera el entorno mediante el ensamblado y repetición de una serie de módulos y/o patrones. El empleo de módulos para la creación de entornos virtuales es una técnica ampliamente difundida en la generación de videojuegos 2D (*tile-based games*) [192]. En este caso los módulos o losetas a repetir son cuadrados o hexágonos que se acoplan en el plano hasta forma una malla 2D, según muestra la siguiente figura.

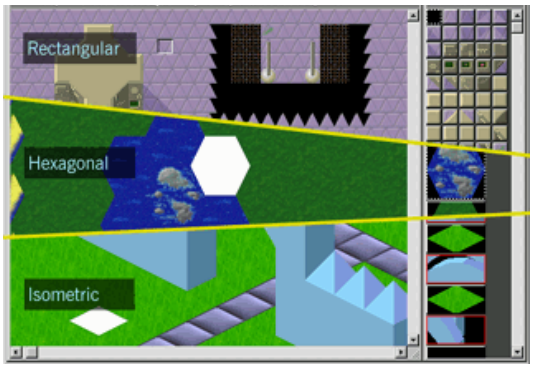


Figura 3.16.Ejemplo de diversas tipos de losetas empleadas en videojuegos 2D.

El efecto 3D se consigue mediante proyección isométrica y superposición de capas. La siguiente figura muestra ejemplos de losetas y su combinación para la generación de un entorno.

[190] Hitchner, L. E. and McGreevy, M. W. Methods for User-Based Reduction of Model Complexity for Virtual Planetary Exploration. In Proceedings of the SPIE, 1993. Vol 1913, pp 622-636.

[191] Funkhauser, T., Sequin, C. Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments. In Proceedings of SIGGRAPH 93, pp 247-254.

[192] Kelly, C. Chapter 10: Tiled games. Programming 2D Games. A K Peters/CRC Press 2012. ISBN: 978-1466508682.

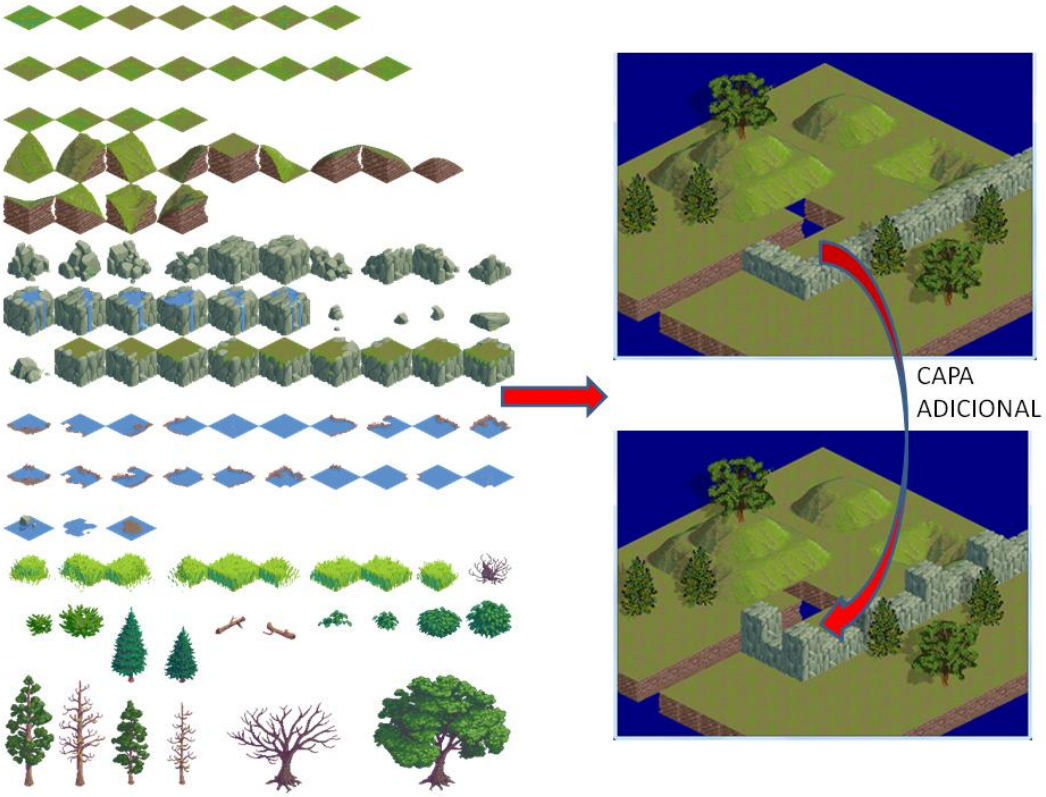


Figura 3.17.Ejemplo de uso de diversas losetas (iso-hexágonos) en la generación de un entorno.

Algunos de estos videojuegos han evolucionado a una combinación de elementos 2d y 3d, donde el terreno se convierte en una malla reticulada que puede ser deformada adoptando diversos relieves. Sobre ella se asientan losetas 2D que contienen diversos elementos (edificios, árboles,etc) y se insertan carreteras, ríos ..etc. La rigidez de este sistema suele venir determinada por las limitaciones de navegación, las restricciones impuestas a la hora de seleccionar los posibles lugares donde asentar una loseta, el elevado número de elementos contenidos en cada patrón, lo que da lugar a la percepción de la repetibilidad de los mismos y la limitación en el tipo de carreteras y trazados ferroviarios modelados. Todo esto restringe enormemente la flexibilidad constructiva de estos entornos.

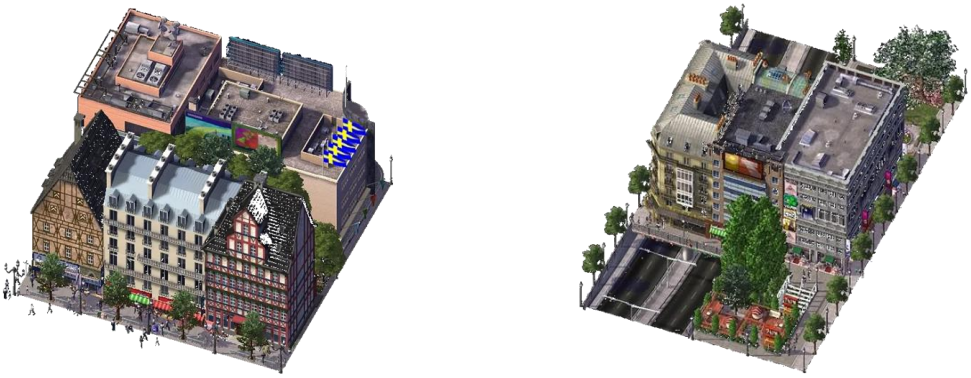


Figura 3.18.Ejemplos de losetas de ciudad del videojuego Simcity [193].

[193] http://www.simcity.com [Última consulta: 9 Enero 2013].

Esta metodología de creación del entorno mediante losetas también ha sido empleada en otros simuladores de conducción, como es el caso de SimVista (Figura 3.19) e ISAT [194], siendo nuevamente la rigidez constructiva la principal limitación de sus escenarios.

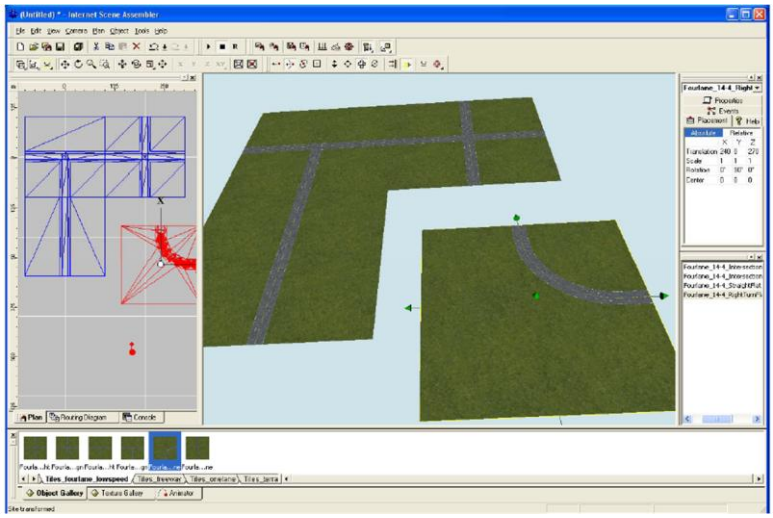


Figura 3.19.Ejemplos de losetas empleadas en SimVista.

Para salvar estas limitaciones la Tecnología Modular genera su propio sistema de módulos aportando las siguientes novedades, que serán explicadas en detalle a lo largo de esta Tesis:

- **La forma de definir el módulo**: el módulo es una porción recta de entorno, que será curvada una serie de grados lo que permitirá reproducir con toda la exactitud necesaria la geometría de la trayectoria y su entorno circundante.
- **El concepto de familia de módulos**: previa búsqueda de unos patrones repetitivos del entorno, tanto geométricos como funcionales, la Tecnología Modular discretiza el escenario en un número finito de familias. La discretización geométrica se realizará en base a perfiles transversales, lo que permitirá una intuitiva parametrización del entorno. La discretización funcional permitirá optimizar el grafo de la escena ya que una misma funcionalidad suele ir acompañada de un mismo tipo de transformaciones geométricas. El hecho de tener agrupados en el grafo de la escena todos los elementos por familias, agilizará dichas transformaciones.
- **El sistema de posicionamiento de módulos**: las trayectorias son en esta tecnología el pilar constructivo, lo que permite por un lado dotarlas de la precisión y continuidad tangencial requeridas y por otro lado aprovechar el conocimiento a priori de estos caminos para generar niveles de detalle discretos pseudo-variantes con el punto de vista. Esto último se consigue con una discretización del entorno transversalmente a la trayectoria.
- **La aplicación de shaders a dichos módulos**: los módulos serán deformados en tiempo real, incrementándose así la flexibilidad constructiva y potenciando aún más las ventajas de la instanciación, disminuyéndose más drásticamente las labores de modelado, la memoria requerida para el almacenamiento de geometrías y los tiempos de carga de la escena.

[194] Fisher D.L, Rizzo M, Caird J.K et al. Chapter 6: Authoring Scenarios. Handbook of Driving Simulation for Engineering, Medicine and Psychology. U.S. CRC Press 2011. ISBN: 978-1-4200-6100-0.

La Tecnología Modular alcanza sus máximos beneficios en las simulaciones de conducción terrestre guiada, si bien, la combinación de los módulos con mallas permite generar siempre todo tipo de entorno con unas elevadas prestaciones.

Como resultado se obtiene un escenario con todo el realismo que las simulaciones de conducción requieren, con las velocidades de refresco deseadas, con una cómoda interfaz de comunicación con el módulo de conducción y motor gráfico y todo ello de una forma automática y con la posibilidad de introducir cualquier reestructuración del entorno fácilmente, algo indispensable en este campo.

A lo largo de este capítulo se explicará en detalle el módulo, primitiva de trabajo de esta metodología y las familias de módulos. La determinación del número y diseño geométrico óptimo de los módulos así como la variedad de familias de éstos serán decisivos para conseguir el realismo y velocidad de refrescos deseados en estas simulaciones.

3.5 EL MÓDULO.

La generación modular basa sus desarrollos en la explotación de una herramienta de optimización clave en el mundo de la realidad virtual: la instanciación. Una instancia es un puntero a un objeto previamente creado y una transformación afín. De esta manera, la instancia reproduce al objeto deformado por su transformación afín[195].

La instanciación es una técnica auxiliar indispensable en situaciones donde la memoria disponible es escasa. Esta búsqueda de patrones repetitivos ha sido el motor de la Tecnología Modular y ha llevado a la distinción dentro de todo entorno virtual entre dos tipos de zonas: zonas modulares y zonas mallables.

Zonas modulares:

Se generan mediante el ensamblado y repetición de un número finito de módulos. Los módulos son porciones de entorno que adecuadamente repetidos, posicionados y jerarquizados dentro de un grafo de la escena, permiten representar virtualmente éste.

El módulo parte de un diseño básico constituido por una porción longitudinal recta de entorno. Este diseño básico se curva una serie de grados, constituyendo lo que se denomina un *conjunto base de elementos modulares* o módulos (Figura 3.20).

De esta forma el conjunto base se caracteriza por la existencia de un número finito de módulos, de curvatura fija y de valores de curvatura discretos.

Son varios los factores a tener en cuenta para conseguir los máximos beneficios de este sistema de instanciación:

- El número óptimo de módulos a emplear: dependerá de sus dimensiones y de las condiciones de repetividad del entorno.
- El diseño geométrico de éstos.
- Las máximas deformaciones que admiten sin sufrir un excesivo deterioro geométrico o de texturado.

[195] Sutherland, I. E. Sketchpad: A man-machine graphical communication system. Proceedings of the Spring Joint Computer Conference, 1963.

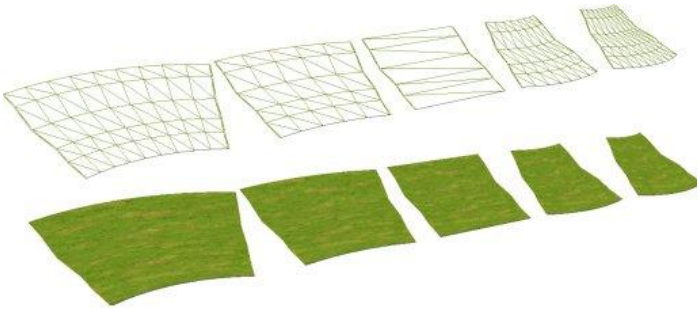


Figura 3.20.Conjunto base de módulos.

Las características que ha de reunir un entorno para poder llevar a cabo una representación modular del mismo son:

- Existencia de una dirección privilegiada, que actuará como línea directriz para el posicionamiento de los módulos.
- Existencia de elementos repetitivos o elementos cuya condición de autosimilitud, permitan sintetizar su generación mediante la combinación de un número finito de fragmentos instanciados. Esta metodología, no pretende generar un modelo idéntico al de partida, sino que crea uno nuevo tal que el usuario de una simulación de conducción terrestre pueda cumplir todos los objetivos de su entrenamiento en un entorno lo más parecido posible al mundo real. Es la llamada instancia aproximada [196] (*approximate instancing*).

Siguiendo esta línea la Tecnología Modular busca la discretización del entorno en un conjunto de perfiles transversales y líneas directrices que actuarán como base de extrusión de los mismos, y que adecuadamente posicionados e instanciados permitirán representar el entorno. La Figura 3.21 muestra un ejemplo de empleo de estos módulos en la construcción de entornos virtuales.

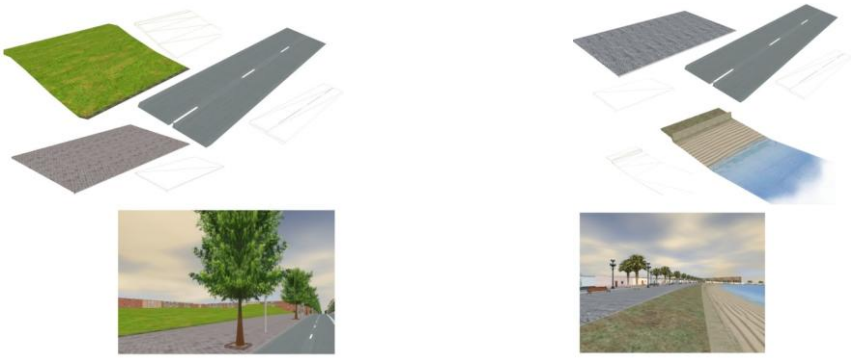


Figura 3.21.Ejemplo de empleo de módulos.

Zonas mallables.

No siempre es posible asimilar un fragmento de entorno a un conjunto de elementos repetidos. La existencia de geometrías irregulares, en las que no es posible determinar una línea directora, impide el empleo de esta tecnología. En estos casos las zonas se modelan mediante las

[196] Lintermann, B. Digital Design of Nature: Computer Generated Plants and Organics. Springer 2005, pp 137. ISBN: 978-3642073632.

clásicas mallas de triángulos. La Figura 3.22 muestra ejemplos de zonas donde los módulos no son aplicables y se hace necesario el empleo de mallas.

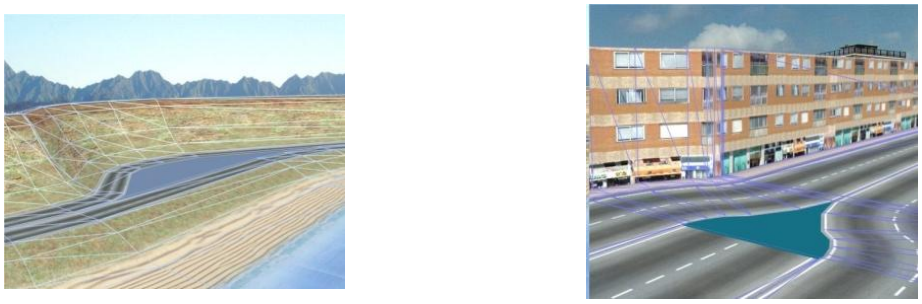


Figura 3.22.Ejemplo de empleo combinado de módulos y mallas.

3.5.1 COMPONENTES DEL MÓDULO.

El módulo está formado por tres elementos básicos: un recorrido longitudinal y un conjunto de perfiles transversales con sus correspondientes texturas (Figura 3.23).

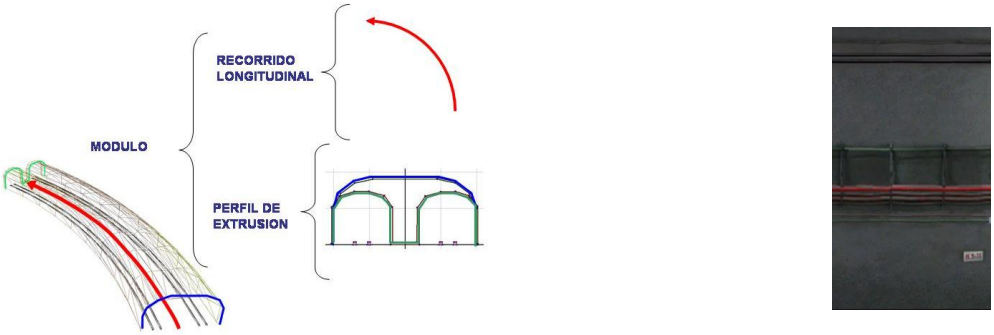


Figura 3.23.Componentes de un módulo: recorrido longitudinal, perfiles transversales y textura.

Estos elementos se encuentran asociados en estructuras denominadas solevados. La extrusión de los perfiles a lo largo del recorrido longitudinal genera la geometría tridimensional del módulo.

Con el fin de hacer el texturado lo más versátil posible, cada perfil transversal se divide en tantos subperfiles, como texturas se deseen colocar. La asociación de cada subperfil, con su textura y recorrido longitudinal constituye un microsolevado. El conjunto de microsolevados transversales definen el módulo o solevado final, como muestra la Figura 3.24.

Figura 3.24.Composición final del módulo.

El proceso constructivo de un módulo comienza con la discretización poligonal de su recorrido longitudinal o *línea directora*, generándose la *poligonal directora*. La Figura 3.25 muestra diferentes grados de discretización poligonal para un conjunto de módulos rectos y curvos. La discretización en función de la curvatura permite una mayor optimización de la carga gráfica.

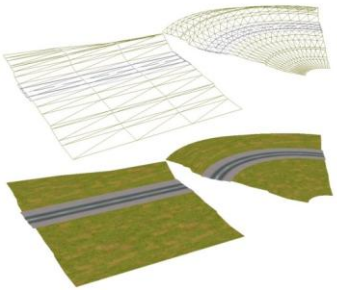


Figura 3.25.Ajuste del número de polígonos según la curvatura.

Para determinar el nivel de discretización óptimo se evalúa el parámetro *ángulo de suavizado*. Este parámetro (Figura 3.26) marca el máximo ángulo que pueden formar las normales de los planos definidos por dos triángulos adyacentes de la misma malla y se define para cada una de las familias.

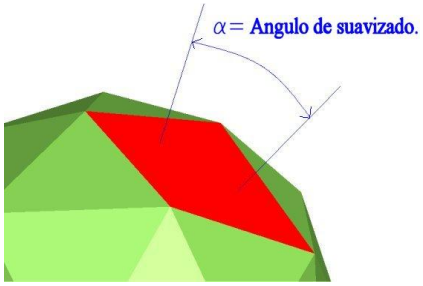


Figura 3.26.Concepto de ángulo máximo de suavizado

Para realizar la extrusión del perfil se determina el plano transversal, en cada uno de los puntos de la línea directora resultantes de la discretización poligonal (sobre los que más tarde se proyecta el perfil). Cada uno de los planos tendrá asociado un parámetro denominado *recorrido relativo de la extrusión*.

Sea el plano P asociado al punto A de la discretización poligonal (Figura 3.27):

- P.R.E.(P) : Parámetro de recorrido del plano P.
- LR(A): Longitud recorrida sobre la poligonal directriz desde el inicio hasta el punto A.
- LT: Longitud total de la poligonal directriz.

$$\text{P.R.E.(P)} = \frac{LR(A)}{LT}$$

Puesto que la línea directora viene determinada por un arco de circunferencia, la dirección normal del plano se calcula simplemente teniendo en cuenta el giro relativo del punto sobre el arco.

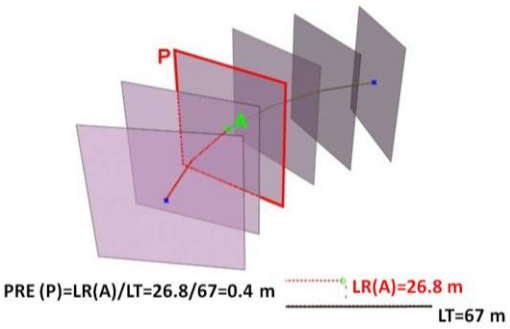


Figura 3.27.Plano transversal en cada punto de la línea directora.

Sobre cada uno de estos planos transversales se proyectará un perfil. El perfil transversal de inicio no tendrá porque coincidir con el final, por tanto se deben calcular perfiles de transición entre ambos para los planos intermedios.

Para ello es necesario buscar pares de puntos análogos en los perfiles de inicio y fin, lo que se denomina *muestreo pareado*. Se presentan dos criterios geométricos para determinar la analogía entre puntos de diferentes perfiles.

- **Criterio de coordenada transversal** (Figura 3.28). Según este criterio para que dos puntos de diferentes perfiles sean análogos la coordenada transversal respecto de la línea directriz debe ser igual.

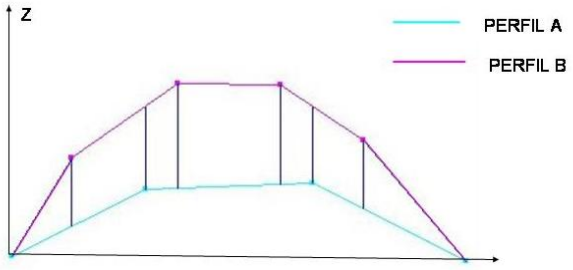


Figura 3.28.Muestreo de perfiles siguiendo el criterio de coordenada transversal.

Este criterio se utiliza para perfiles en los que no existen dos puntos que compartan la misma coordenada transversal. Este es el caso de módulos de carreteras, vías de tren, etc.

- **Criterio de monotonía** (Figura 3.29). Dos puntos son análogos si tienen la misma distancia relativa recorrida sobre sus respectivas poligonales. Es decir, para un par de perfiles de inicio y fin, dados por dos poligonales A y B respectivamente, un punto Pa de A es análogo de otro punto Pb de B si se cumple:

$$\text{DRA(Pa)} = \frac{D(Pa)}{D(A)} = \frac{D(Pb)}{D(B)} = \text{DRB(Pb)}$$

Donde se define:

  - D(Pa) = Distancia recorrida a través de la poligonal A desde el inicio de ésta hasta el punto Pa.
  - D(A) = Longitud o distancia total recorrida a través de la poligonal A.
  - DRA(Pa) = Distancia recorrida relativa del punto Pa sobre A.

- D(Pb) = Distancia recorrida a través de la poligonal B desde el inicio de ésta hasta el punto Pb.
- D(B) = Longitud o distancia total recorrida a través de la poligonal B.
- DRB(Pb) = Distancia recorrida relativa del punto Pb sobre B.

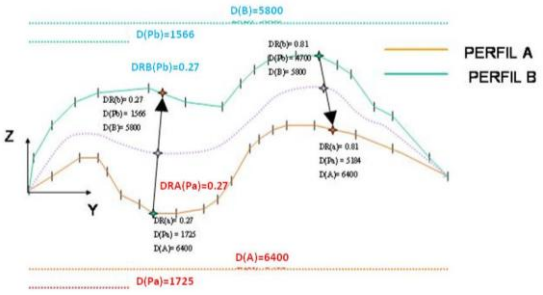


Figura 3.29.Muestreo de perfiles siguiendo el criterio de monotonía.

Este criterio se utiliza para perfiles en los que existen puntos que comparten la misma coordenada transversal. Este es el caso de módulos de túneles, pasillos, etc.

Se define el parámetro *k* (Figura 3.30) como el tanto por uno de longitud recorrida sobre la curva de interpolación que une puntos análogos en los perfiles de inicio y fin.

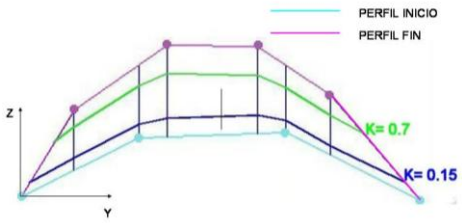


Figura 3.30.Asignación del parámetro K a los perfiles intermedios.

Esta curva presenta tangente nula en los extremos para garantizar siempre una transición suave entre módulos (Figura 3.31). Por ejemplo, en el caso de emplear una curva polinómica de tercer grado, se establece una relación biyectiva entre el parámetro de recorrido relativo de extrusión (P.R.E) y el valor de *k* asociados a un determinado plano transversal como sigue:

K(P) = parámetro *k* asociado al perfil P

$$K(P) = PRE(P)^2 \cdot (3 - 2 \cdot PRE(P))$$

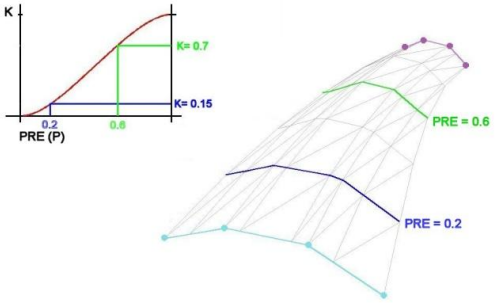


Figura 3.31.Asignación de perfiles intermedios.

A cada plano transversal se le asignará un valor del parámetro *k* y un perfil de transición. La Figura 3.32 muestra las diferencias resultantes de emplear una curva polinómica de tercer y primer grado en la definición del parámetro *k* .

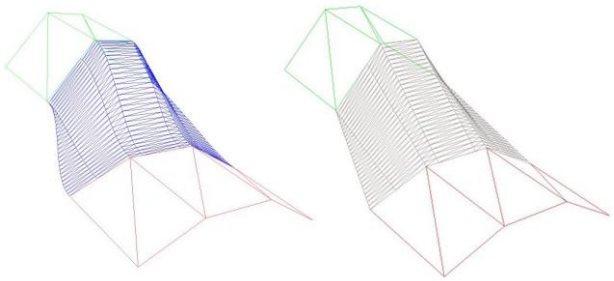


Figura 3.32.Ejemplo de transición suavizada frente a transición lineal

La proyección de cada uno de los perfiles asociados a cada plano transversal sobre dichos planos, genera las secciones del módulo. Para realizar la proyección se calcula en cada uno de los puntos de la línea directriz la terna de vectores unitarios intrínseca como muestra la Figura 3.33.

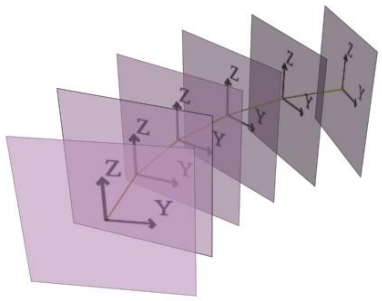


Figura 3.33.Terna intrínseca en cada punto de la línea directora.

Se realiza este proceso con todos y cada uno de los perfiles de los tramos de la configuración transversal obteniendo así la estructura base del módulo, según indica la Figura 3.34.

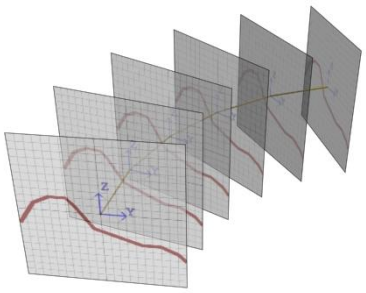


Figura 3.34.Proyección de los perfiles.

La construcción de la malla tridimensional que constituye el módulo se realizará en base a parejas de triángulos que formen paralelogramos (*quads*), para evitar en lo posible la deformación de la textura. Para ello, se recorre el módulo longitudinalmente y por cada par de secciones transversales, se realiza un recorrido transversal de sus perfiles, tomando nuevamente de dos en dos los puntos de cada una de las secciones. Con estas dos parejas se forma un rectángulo (*quad*).

El orden en que son tomados los puntos del quad para generar los triángulos no es aleatorio, sino que este orden es el que determina la normal visible del objeto (Figura 3.35).

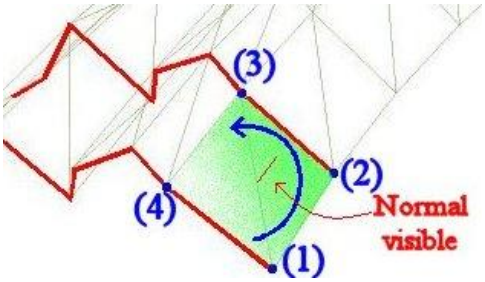


Figura 3.35.Definición de la orientación visible del quad.

Haciendo el recorrido por todas las secciones y perfiles se generará toda la malla. La Figura 3.36 muestra un ejemplo de malla. En verde aparece sombreado un quad.

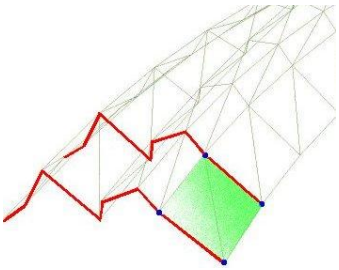


Figura 3.36.Proceso de obtención de quads.

Cada microsolevado lleva asociado una textura. El mapeado de dicha textura se realizará en base a las coordenadas transversal y longitudinal de cada uno de los puntos que definen cada microsolevado (Figura 3.37). La coordenada de textura longitudinal, vendrá determinada por el parámetro de recorrido *k* de cada punto y el parámetro de *tiling longitudinal* de la textura (número de veces que se repite la textura por unidad longitudinal recorrida). Así pues todos los puntos de una misma sección tendrán asignada la misma coordenada de textura longitudinal, garantizando así que la textura queda colocada coherentemente con la monotonía impuesta por la línea directriz. De la misma manera la coordenada de textura transversal vendrá determinada por el parámetro de recorrido transversal de cada punto y el *tiling trasnsversal*.

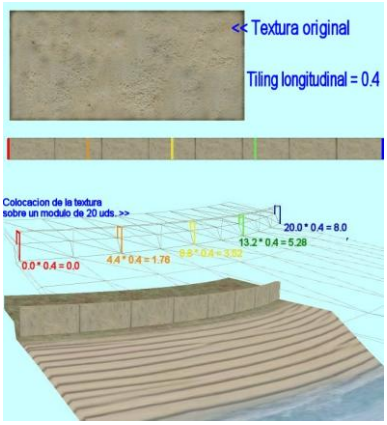


Figura 3.37.Colocación de la textura sobre la malla.

Un elemento del módulo a comentar son las solapas. Dado que los entornos a representar poseen curvaturas variables, cambios de pendiente y peralte que deben ser cubiertos por un número limitado de elementos que han sido curvados únicamente en planta (*conjunto base de módulos*), será necesario tener esto en cuenta en el diseño geométrico del módulo. Para ello los módulos se generan con solapas en sus extremos cuya finalidad es cubrir errores angulares de solapamiento o aperturas entre módulos. Estas solapas se construyen añadiendo a los extremos del módulo dos perfiles transversales que poseen una ligera inclinación con el fin de evitar efectos de Z buffer (Figura 3.38).

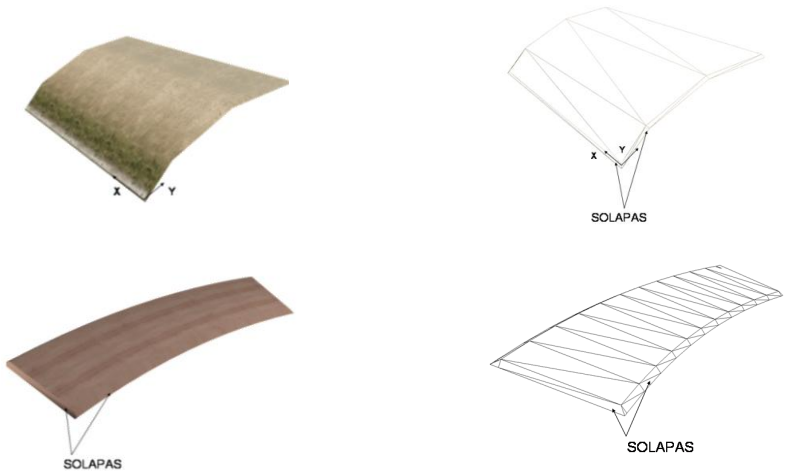


Figura 3.38.Ejemplos de solapas en módulos.

A la hora de diseñar las solapas es necesario tener en cuenta una serie de limitaciones en su uso y tamaño. La primera es que solapas de gran tamaño aumentan el índice de complejidad del repintado en la generación de la imagen. La segunda es que se pueden notar visualmente las uniones entre módulos si no hay un adecuado diseño de las texturas. La tercera es el problema de parpadeos por la superposición de polígonos paralelos y cercanos, así como problemas de iluminación que exigen una selección adecuada de los ángulos de inclinación de las solapas.

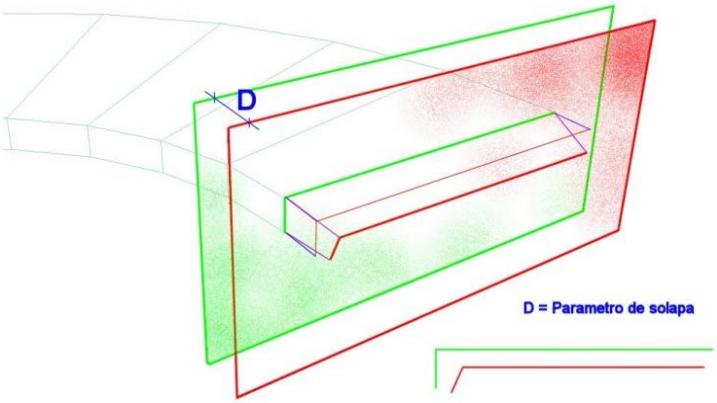


Figura 3.39.Construcción de la solapa de concatenación para el modulo.

A continuación se muestran un ejemplo de entorno creado sin y con solapas (con y sin fisuras).

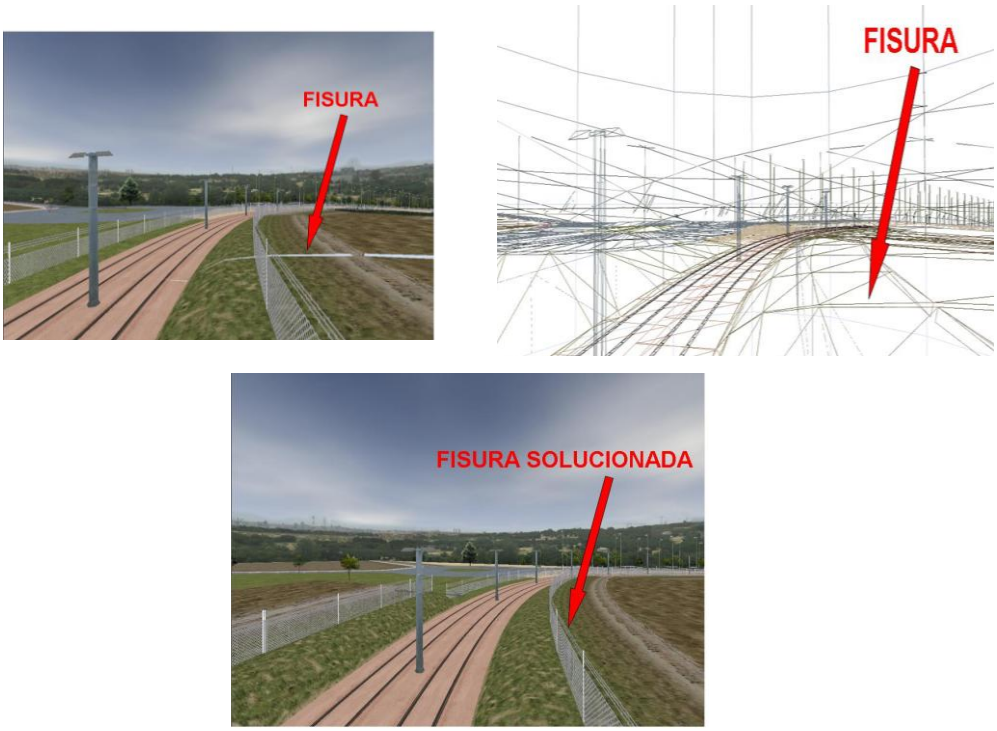


Figura 3.40.Ejemplo de entorno generado sin y con solapas.

Sin embargo, el empleo de shaders, como se explicará en el Capítulo 5 permite la deformación en tiempo real del módulo, adaptándolo a la forma geométrica final que se desee, evitando así la introducción de solapas y sus posibles inconvenientes.

Finalmente a la hora de diseñar un módulo hay que tener en cuenta una serie de aspectos básicos:

- el empleo de un formato entendible por el motor gráfico.
- estos módulos han de seguir una codificación en su nomenclatura con el fin de que cada uno de ellos sea único en la representación virtual y sea posible su identificación y jerarquización. En el Anexo 1, se explica la nomenclatura definida por la Tecnología Modular.

3.6 LA FAMILIA DE MÓDULOS.

A la hora de representar modularmente un entorno, se ha visto como es necesario definir en primer lugar un *conjunto base de módulos*, tales que, su diseño geométrico, permita mediante una adecuada algorítmica de posicionamiento, reproducir cualquier geometría sin agujeros ni solapamientos. El problema siguiente se plantea al pretender determinar los perfiles transversales de dichos módulos, para que puedan reproducir cualquier entorno. Surge así el concepto de *familia* de módulos, como el conjunto base de módulos, que comparten una misma definición de perfiles transversales (Figura 3.41). Estas familias verificarán unas relaciones de compatibilidad, que serán las que garanticen la ausencia de incoherencias o discontinuidades tras el correcto posicionamiento de los módulos.

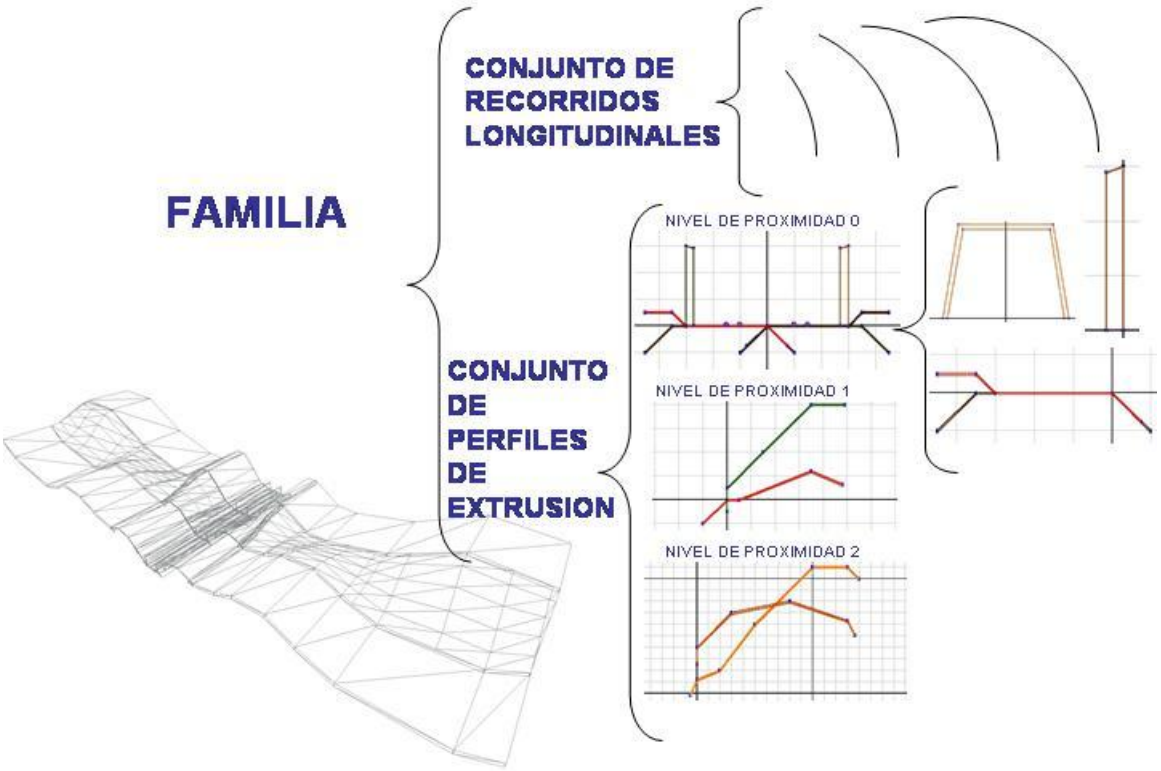


Figura 3.41.Componentes de una familia de módulos.

La definición del conjunto de familias que definen un determinado entorno virtual, hace necesario tener en cuenta en primer lugar una discretización funcional del mismo, que haga referencia tanto a su papel dentro del entorno físico como del entorno gráfico. De esta manera en esta Tesis se distinguirá en primer lugar entre familias de terreno, que a su vez se dividirán en familias para la representación de trayectorias circulatorias, y familias de terreno circundante y familias de los restantes elementos posicionados sobre éste.

Para definir las familias, será necesario tener en cuenta las diversas operaciones gráficas a las que podrán verse sometidas los diferentes elementos que constituyan un módulo. Se buscará que todos los módulos pertenecientes a una familia compartan las mismas operaciones gráficas (como la iluminación) con el fin de mantener agrupados dichos elementos en el grafo de la escena y agilizar así los cálculos que ha de llevar a cabo el motor gráfico.

3.6.1 FAMILIAS PARA LA REPRESENTACIÓN DE TRAYECTORIAS CIRCULATORIAS.

Las trayectorias circulatorias son el pilar constructivo de la Tecnología Modular. Por un lado guían el tráfico vehicular y por otro lado actúan como líneas directrices para la inserción modular. Una vez definida la topología y geometría de estas trayectorias, se posicionarán los módulos a partir de las mismas (Capítulo 4), siguiendo para ello los Algoritmos de Posicionamiento Modular (Capítulo 5).

Ejemplos de este tipo de familias son las plataformas de tren, las traviesas y las carreteras. La Figura 3.42 muestra ejemplos de este tipo de módulos.

Figura 3.42.Ejemplos de módulos para modelar las trayectorias circulatorias.

La Figura 3.43 muestra algunos ejemplos de aplicación de la Tecnología Modular en proyectos llevados a cabo por CITEF para Metro de Madrid. En primer lugar se detallan las líneas directrices que servirán de posicionamiento de módulos. A continuación se muestra el entorno 3d generado con los módulos, y finalmente se expone una visión alámbrica de los mismos que permite distinguir la concatenación de módulos empleada.

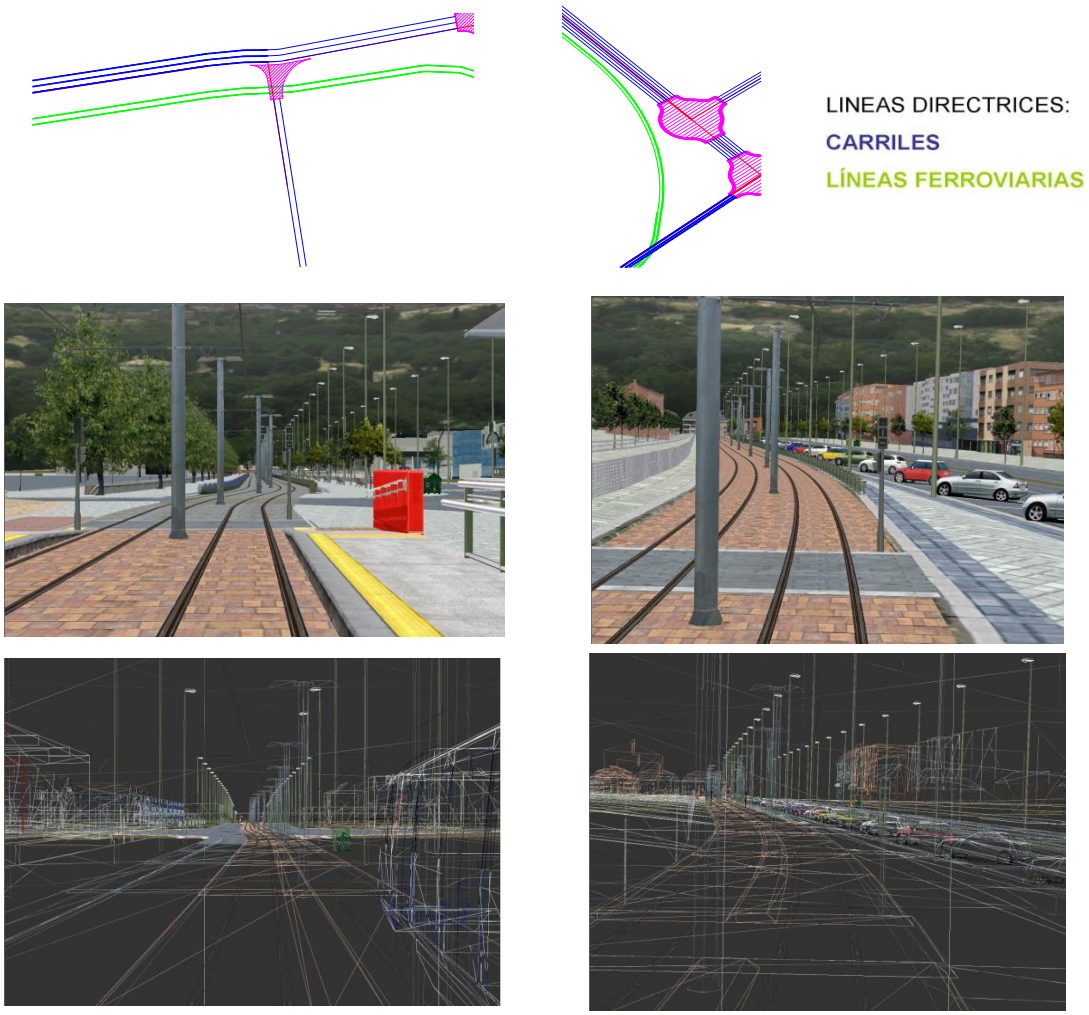


Figura 3.43.Ejemplos de empleo de módulos para representar las trayectorias circulatorias.

Un elemento muy singular que requiere un tratamiento especial a la hora de emplear módulos en su generación es la aguja. Las agujas se modelan a partir de tres tipos de módulos: los carriles, las solapas y los espadines.

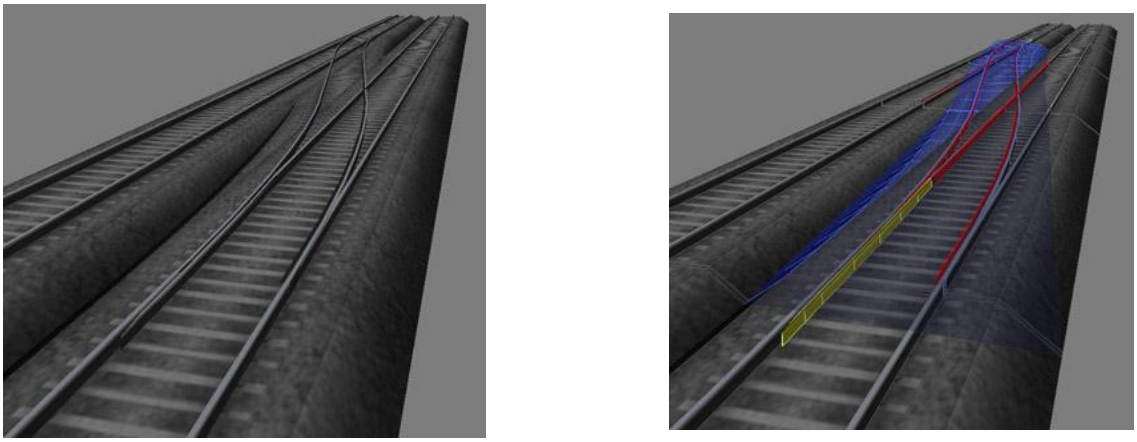


Figura 3.44.Solapas en rojo, espadín en amarillo.

Las solapas son empleadas para emular huecos y poder reproducir así el corazón de la aguja y el movimiento del espadín. Existen dos tipos de solapas: fijas y móviles. Las solapas fijas se mantienen estáticas a lo largo de toda la simulación, mientras que las móviles aparecen y desaparecen (mediante un switch en el grafo de la escena) en función de que el estado de la aguja sea directo o desviado.

El espadín es el elemento gracias al cuál, el tren puede realizar el cambio de vía garantizando siempre la continuidad del raíl, evitando así el descarrilamiento.

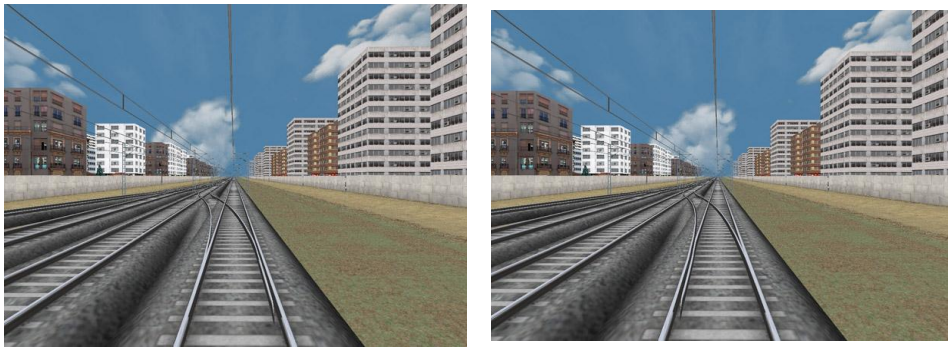


Figura 3.45.Aguja en estado directo y desviado.

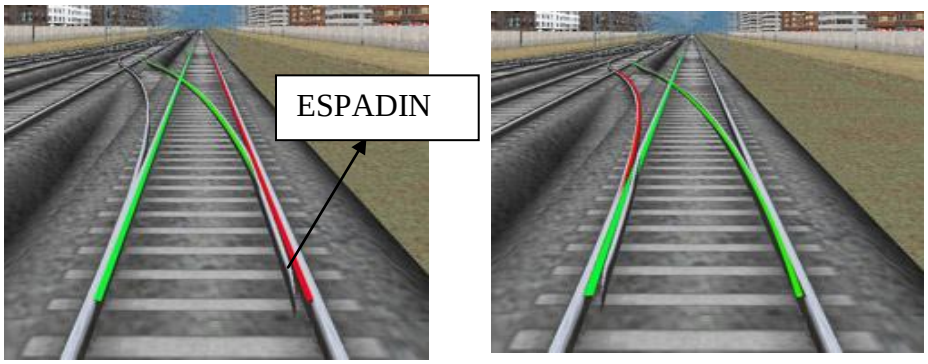


Figura 3.46.Solapas móviles en rojo y solapas fijas en verde para la aguja en estado directo y desviado respectivamente.

3.6.2 FAMILIAS DE TERRENO CIRCUNDANTE.

Cuando se desea representar virtualmente un terreno generalmente se da la circunstancia de que su magnitud longitudinal, siguiendo la línea directriz del entorno, supera las dimensiones básicas de los elementos modulares, por lo que es necesario aplicar algoritmos que generen paisajes, discretizando para ello previamente estos entornos en familias de módulos de terreno. La Figura 3.47 y Figura 3.48 representan de manera esquemática esta situación.

Figura 3.47.Entorno formado por la definición de un paisaje "Colina".

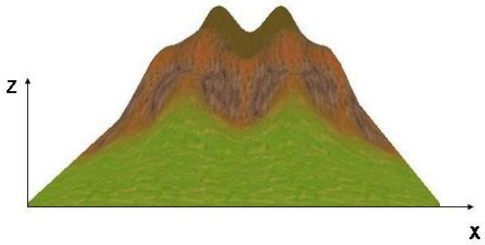


Figura 3.48.Paisaje "Colina" definido en un intervalo más pequeño.

Los elementos modulares que componen una familia de terreno están subdivididos según la distancia transversal a la línea directriz del entorno y la disposición lateral (izquierda, derecha o centrados). Se forman así lo que se denominan niveles de proximidad (Figura 3.49). Con esta subdivisión se buscan dos objetivos. En primer lugar reducir la carga sobre el visualizador haciendo los elementos mas alejados con un menor grado de detalle. En segundo lugar maximizar la modularidad del entorno para conseguir el mayor número de configuraciones posibles con el menor número de módulos.

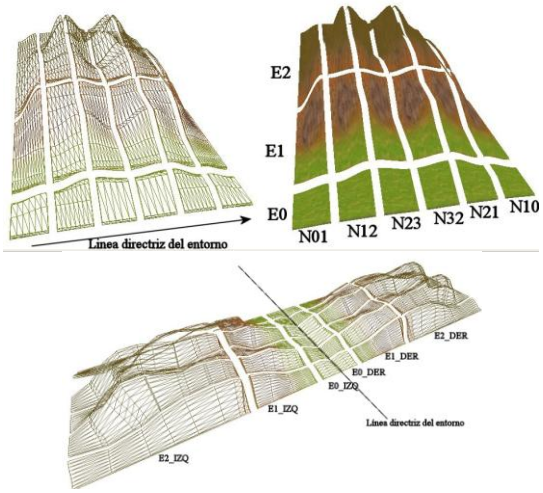


Figura 3.49.Distribución de la familia Terreno según entornos de proximidad y posicionamiento lateral.

Una familia de terreno quedará por tanto definida a partir de la sintetización del entorno en un conjunto de perfiles transversales, sus posibles texturas y la definición de una curva de transición entre dichos perfiles. Para poder representar satisfactoriamente el entorno deseado, es necesario concatenar estos módulos de forma adecuada. Esto se lleva a cabo mediante el concepto de *paisaje*.

Las características que distinguen un paisaje de terreno de otra serán:

- Sinuosidad del terreno.
- Pendiente.
- Coordenada z máxima y mínima.
- La textura.

Estos parámetros quedan sintetizados en una determinada secuencia de perfiles transversales.

En función del tipo de entorno a representar, real o ficticio, y la disponibilidad de información en el primer caso, la generación de la familia de Terreno y los paisajes estará más o menos automatizada. Su generación será descrita en el Capítulo 4.

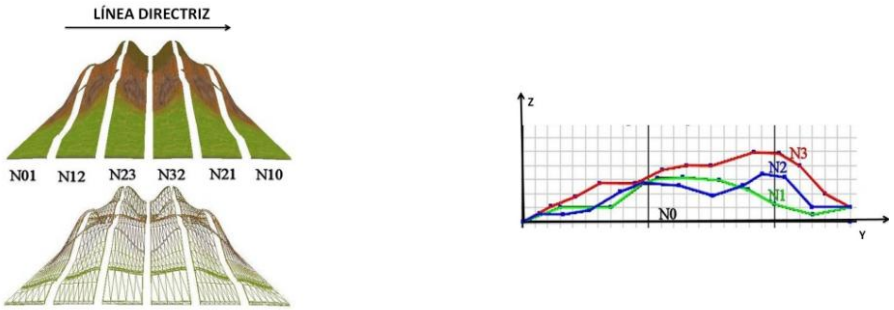


Figura 3.50.Paisaje colina con la secuencia de módulos que la definen. Definición de los perfiles transversales empleados.

Dentro de este grupo de familias también se engloba todo aquél terreno circundante a una trayectoria circulatoria, es decir, cualquier zona pavimentada o sin pavimentar sobre la que no exista tráfico vehicular, como pueden ser las aceras, playas, etc. A continuación se muestran una serie de ejemplos de entornos creados con el uso de módulos (Figura 3.51)

Figura 3.51.Ejemplos de empleo de módulos para representar el terreno circundante a las trayectorias circulatorias.

Un caso especial de familias de terreno circundante son los túneles. Para conseguir una mayor versatilidad constructiva y una máxima instanciación su punto de inserción se coloca en el techo. Los túneles tienen una altura igual a la máxima encontrada en el entorno, de manera que puede conseguirse el efecto de una altura variable de túnel, calculando correctamente la cota z de inserción del módulo de túnel a partir de la cota z de la línea directriz y de la altura de túnel deseada en cada punto kilométrico.

Figura 3.52.Módulos de túnel.

3.7 ZONAS MALLABLES

El tratamiento que se dará a cada zona mallable o singularidad no será el mismo sino que dependerá de la existencia o no de direcciones privilegiadas que marquen un sentido para el texturado y para la adición de perfiles transversales. Atendiendo a este criterio las singularidades se clasifican en tres grupos:

1.Sin direcciones privilegiadas

Este tipo de singularidades se da cuando no existe una dirección privilegiada en el entorno o existe una anisotropía no uniforme, es decir, no existe una dirección de texturado a seguir. El caso de las intersecciones entre varios carriles en los simuladores de conducción urbana es un ejemplo de este tipo de singularidades (Figura 3.53).

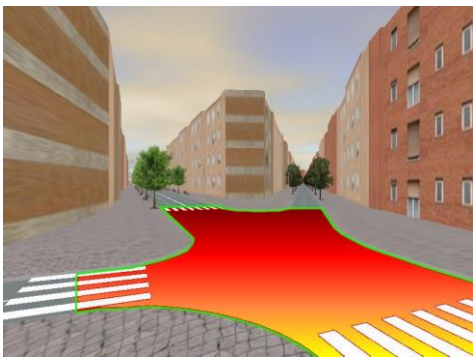


Figura 3.53.Ejemplos de singularidad sin dirección privilegiada: intersección en ciudad.

La malla se generará mediante una triangulación de Delaunay y se aplicará un mapeado homogéneo e isótropo. Para ello se calcula el rectángulo envolvente de la malla y se determinan las coordenadas de textura de cada uno de sus vértices (Figura 3.54).

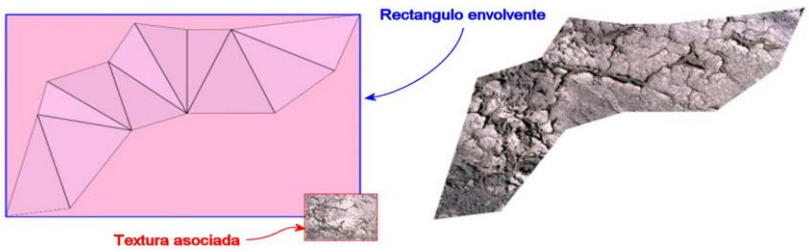


Figura 3.54.Mapeado de la textura sobre una singularidad sin direcciones privilegiadas.

Las coordenadas de textura definitivas son calculadas corrigiendo las anteriores con los parámetros de tileado propios de la textura asignada (Figura 3.55).

Figura 3.55.Ejemplo de malla generada para una singularidad sin dirección privilegiada.

2.Única dirección privilegiada.

Este tipo de singularidades se da en entornos en los que existe una única dirección privilegiada que rige tanto la navegación a través del entorno como su generación. En los simuladores ferroviarios el entorno está claramente construido a partir de la dirección privilegiada que da el eje de la vía y en los de conducción en zonas interurbanas el entorno está estructurado siguiendo la línea de la carretera. Algunos ejemplos de estas singularidades son separación de vías, bifurcaciones e incorporaciones de carril, etc (Figura 3.56).

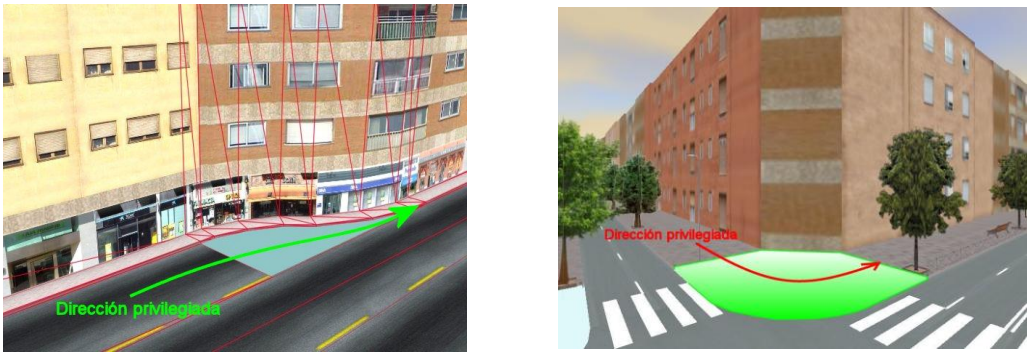


Figura 3.56.Ejemplos de singularidades con una única dirección privilegiada.

El proceso de generación de la zona mallable se estructura en las siguientes fases:

- Construcción de una línea directriz acorde con la dirección privilegiada.
- Generación de una malla orientada que cubra la singularidad según se recorre la línea directriz.
- Adición de los perfiles transversales que permitan reproducir la topografía del entorno.
- Mapeado de la textura.

La línea directriz se obtiene calculando el punto medio de pares análogos en los contornos límites de la singularidad. El criterio para definir si dos puntos son análogos es el Criterio de monotonía definido en el apartado 3.5.1. La poligonal así obtenida se suaviza mediante un proceso iterativo en el que se van tomando ternas sucesivas de puntos. Cada una de estas ternas define un triángulo del cual se calcula su centro geométrico. La nueva línea directriz se generará mediante la unión de dichos centros geométricos. Este proceso iterativo finaliza al alcanzar el grado de suavizado deseado (Figura 3.57).

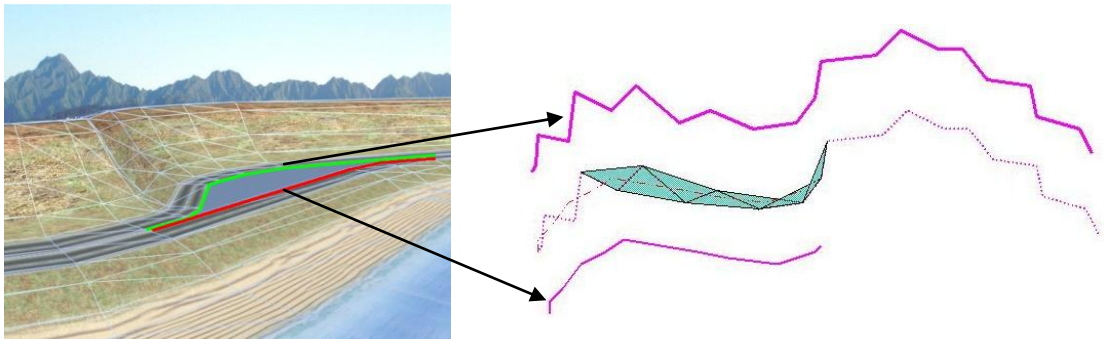


Figura 3.57.Singularidad formada por la separación de dos carreteras generadas por módulos: suavizado de la línea directriz mediante triangulación.

Una vez obtenida la línea directriz que marca la dirección privilegiada a lo largo de la singularidad, se trata de construir una malla que cubra el recinto delimitado por las dos poligonales de contorno. Esta malla debe garantizar la orientación de los triángulos a lo largo de dicha directriz con el fin de permitir la posterior colocación de la textura y la adición de perfiles transversales a lo largo de la misma.

Se comienza así seccionando el recinto en zonas rectangulares (quads) orientadas según la línea directriz. Para llevar a cabo este seccionamiento se recorren ambos contornos buscando puntos análogos en una y otra poligonal según el Criterio de puntos enfrentados, el cual considera que dos puntos son análogos si comparten el mismo plano perpendicular a la línea directriz (Figura 3.58).

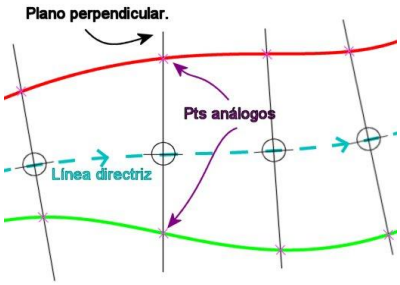


Figura 3.58.Criterio de puntos enfrentados

Para determinar las coordenadas de textura longitudinal (coordenada siguiendo la dirección de la línea directriz) de cada pareja de puntos análogos M y M' se calcula el punto de corte de su sección transversal con la línea directriz, P_C (Figura 3.59). A continuación se determina la distancia relativa recorrida por el punto P_C sobre la línea directriz DR_{LD}:

$DR_{LD}(Pc) = \frac{D_{LD}(Pc)}{D_{LD}}$, siendo $D_{LD}(Pc)$ la distancia recorrida sobre la línea directriz hasta el punto Pc y D_{LD} la longitud de la línea directriz.

Esta variable multiplicada por el factor de tileado de la textura será la coordenada longitudinal.

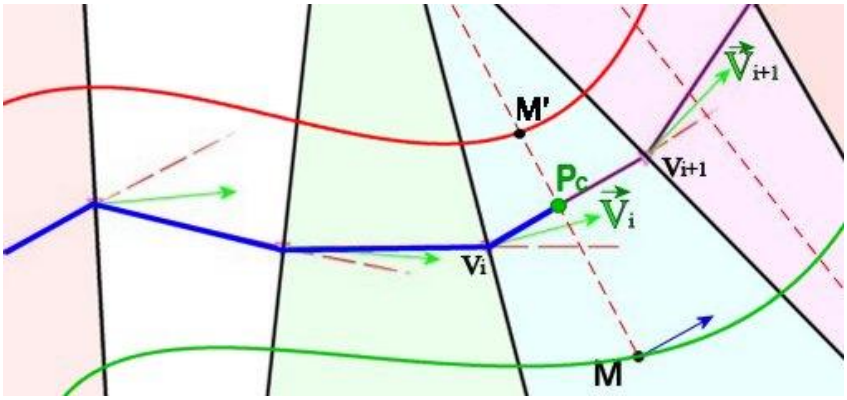


Figura 3.59.Esquema del algoritmo de seccionamiento

Con el fin de evitar quads degenerados debido a la intersección de secciones transversales en trazados con cambios bruscos de dirección, los pares de puntos análogos se reordenan (Figura 3.60).

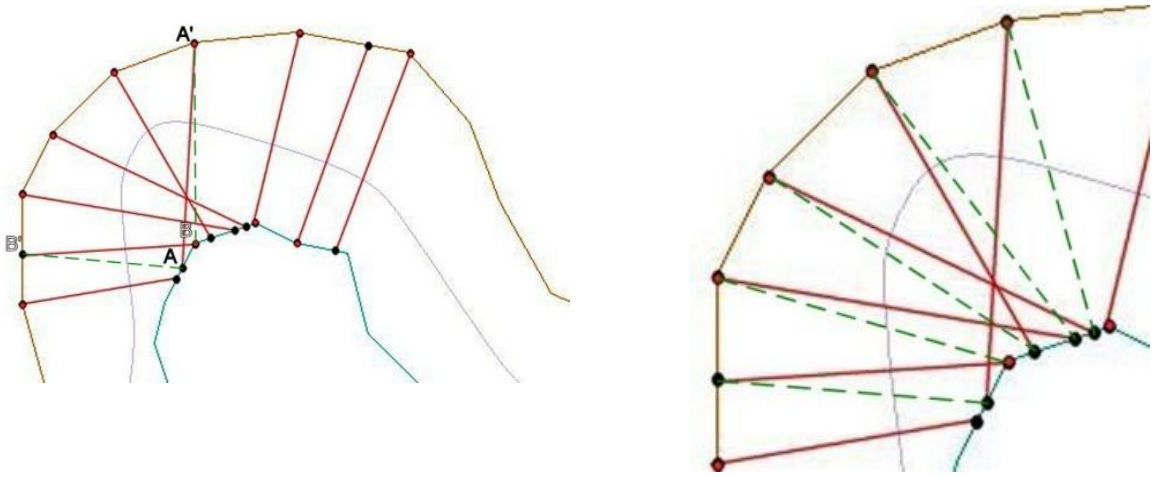


Figura 3.60.Quads antes y después de la reordenación de pares de puntos análogos.

A continuación se incorpora la definición de perfiles transversales. Cada perfil vendrá dado sobre una escala unitaria en la dirección transversal Este perfil será escalado según las necesidades del contorno y proyectado ortogonalmente en la sección transversal correspondiente.

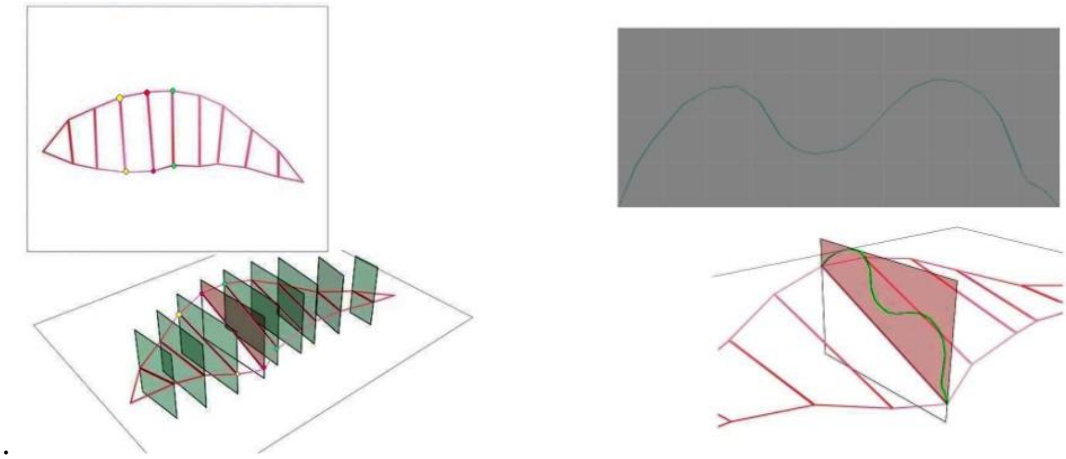


Figura 3.61.Perfil unitario al que se le ha aplicado una transformación de escalado y giro para proyectarlo sobre la sección transversal correspondiente.

Posteriormente se recorre en sentido longitudinal la línea directriz, tomando los perfiles transversales de dos en dos. Cada pareja de perfiles se recorre transversalmente tomando nuevamente de dos en dos los puntos de cada una de las poligonales que definen la sección transversal formando quads (según se vió en la Figura 3.35).

Finalmente se asignará una textura a la geometría. Para el cálculo de la coordenada de textura transversal se considerará por defecto una coordenada 0.0 para aquéllos puntos situados sobre el contorno izquierdo de la singularidad y una coordenada de 1.0 para los que se encuentren sobre el contorno derecho. Estas coordenadas pueden venir afectadas por parámetros de proporcionalidad según el tileado transversal de la textura.

Figura 3.62.Ejemplos de mallas generadas para singularidades con una única dirección privilegiada.

3.Múltiples direcciones privilegiadas

En el caso de multiplicidad de direcciones privilegiadas se trata de encontrar una línea directriz que aglomere la acción de todas las direcciones privilegiadas y aplicar los mismos pasos que en el caso anterior (Figura 3.63).

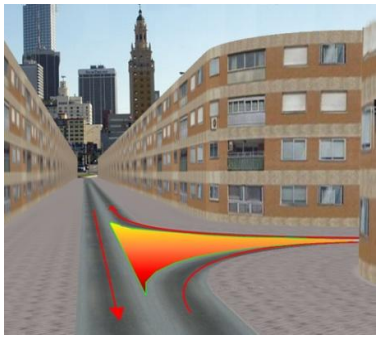


Figura 3.63.Isleta: ejemplo de singularidad con múltiples direcciones privilegiadas.

Para realizar este cometido se busca el árbol de recorrido mínimo de la singularidad a tratar. Dicho recorrido definirá la línea directriz buscada.

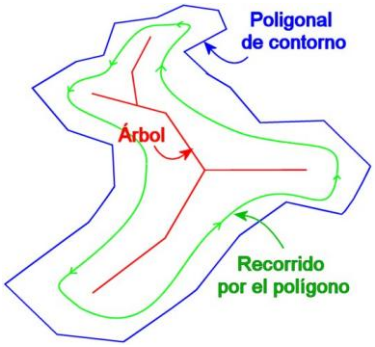


Figura 3.64.Singularidad con múltiples direcciones privilegiadas: árbol y recorrido generado por el mismo.

El árbol de recorrido mínimo para un polígono P es un grafo derivado de la triangulación de Delaunay [197]. De esta manera, basándose en la triangulación de Delaunay se construye el árbol de recorrido de la singularidad mediante la unión de los puntos medios de las aristas internas de la malla.

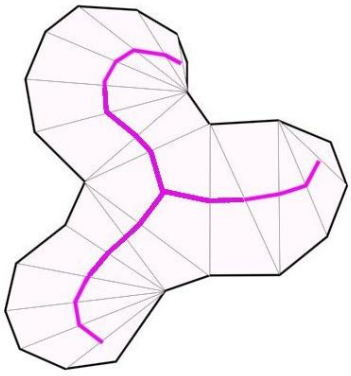


Figura 3.65.Árbol de recorrido derivado de una triangulación de Delaunay.

A continuación con el fin de simplificar al máximo el árbol se eliminan las ramas insignificantes. El algoritmo empleado para determinar y eliminar las ramas insignificantes es

[197] Mount, David M. Lecture 17: Delaunay Triangulations. CMSC 754 Computational Geometry, Department of Computer Science, University of Maryland, Fall 2002. .

utilizado ampliamente en programas para generación de elementos 3D por trazado libre [198] [199], aunque nunca había sido utilizado con el propósito de obtener una línea directriz tal y como se expone a continuación.

Partiendo de la triangulación de Delaunay, se clasifican los triángulos resultantes de esta triangulación como:

- Triángulos que contienen dos aristas externas (triángulos T).
- Triángulos que contienen una arista externa (triángulos S).
- Triángulos con todas sus aristas internas (triángulos J).

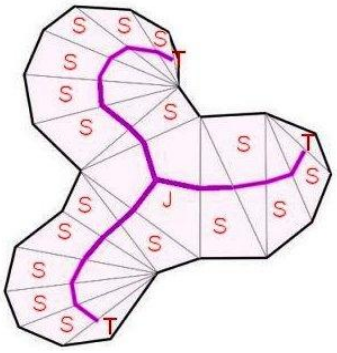


Figura 3.66.Tipos de triángulos en una triangulación de Delaunay.

Sea X un triángulo T, entonces se construye un semicírculo cuyo diámetro es la arista interior del triángulo T. Si los tres vértices de X se encuentran en o dentro de este semicírculo, se elimina la arista y se fusiona X con el triángulo que se encuentra en el otro lado de la arista. Si el triángulo adyacente nuevo es un triángulo S, X tiene ahora tres aristas exteriores y una nueva arista interior. De nuevo se construye un semicírculo en la cara interior y se comprueba que todos los vértices estén dentro de él. Se continúa el algoritmo hasta que algún vértice descanse fuera del semicírculo o hasta que el nuevo triángulo adyacente sea un triángulo J. En el caso en el que algún vértice descanse fuera del semicírculo, se triangula X con un abanico de triángulos que irradian en el punto medio de la arista interior; sin embargo, si el nuevo triángulo adyacente es un triángulo J, se triangula X con un abanico desde el punto medio del triángulo J

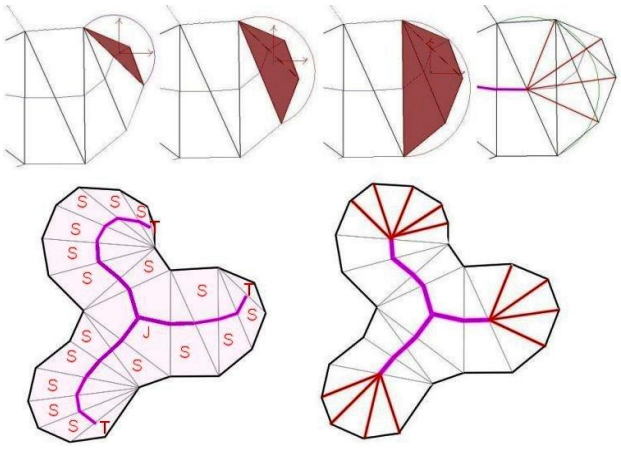


Figura 3.67.Aplicación del algoritmo de eliminación de ramas.

[198] Prasad, L. Morphological analysis of shapes. *CNLS Newsletter*, July 1997. N. 139, pp: 1-18.

[199] Igarashi,T, Matsuoka, S., Tanaka,H. Teddy: A Sketching Interface for 3D Freeform Design. In Proceedings of the 26th annual conference on Computer graphics and interactive techniques, July 1999. ACM SIGGRAPH 99, pp.409-416.

Una vez llevado a cabo el algoritmo de eliminación de ramas no significativas se observa cómo el árbol obtenido permite realizar un recorrido a través de todo el recinto de la singularidad marcando una línea directriz coherente con el contorno y por tanto con las direcciones privilegiadas del entorno.

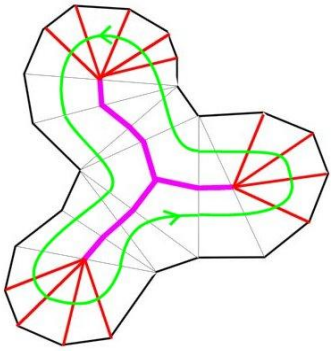


Figura 3.68.Línea directriz (verde) que aglomera las múltiples direcciones de la singularidad.

Siguiendo el recorrido impuesto por la línea directriz y apoyándose en el contorno de la singularidad y en el árbol de recorrido se construye una malla de triángulos que cubre toda la singularidad.

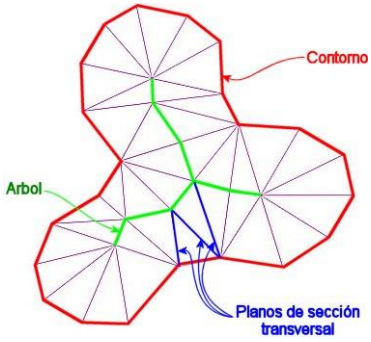


Figura 3.69.Malla de triángulos orientada.

Cada uno de los segmentos que unen los puntos del contorno con los puntos del árbol define un plano proyectante que se denomina plano de sección transversal. Los perfiles vendrán dados sobre una escala unitaria en la dirección transversal y se proyectarán sobre cada sección transversal realizando previamente una transformación de escalado y giro para que encaje sobre el contorno de la singularidad.

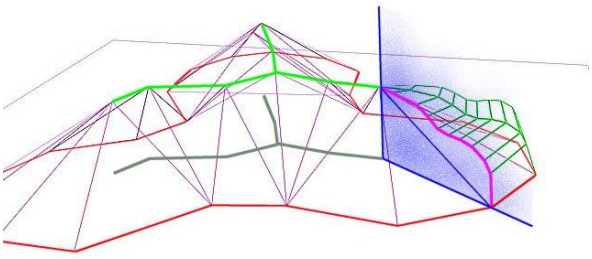


Figura 3.70.Proyección del perfil de entorno.

Finalmente se mapea la geometría. Dado que las texturas a emplear son generalmente rectangulares, la malla se somete a un proceso de rectangularización, convirtiendo la sucesión de

triángulos en una sucesión de quads. Para ello se subdivide la malla en los tramos correspondientes entre dos triángulos T (Figura 3.71).

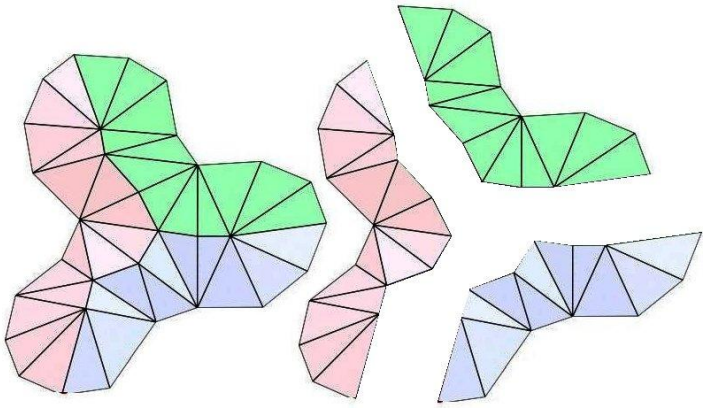


Figura 3.71.Subdivisión de la malla.

Cada una de estas submallas será tratada independientemente como un caso particular de singularidad con una única dirección privilegiada como se vio anteriormente (Figura 3.72).

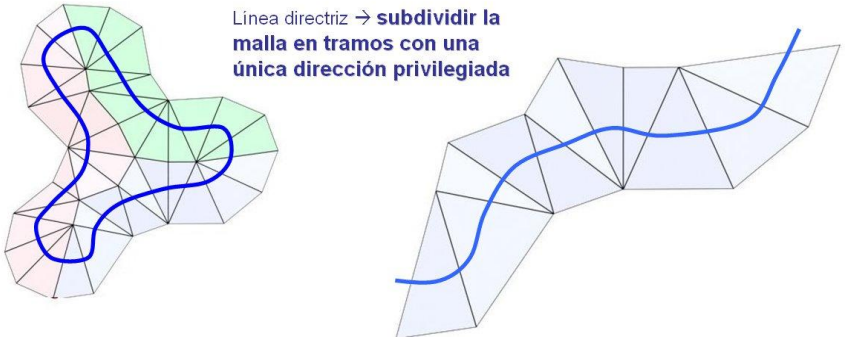


Figura 3.72.Solución basada en la subdivisión en tramos unidireccionales.

La siguiente figura muestra un ejemplo de singualridad multidireccional.

Figura 3.73.Isleta: ejemplo de singularidad multidireccional.

4.LA TECNOLOGÍA MODULAR II: MODELADO DE DATOS.

4.1 INTRODUCCIÓN.

Como ya se comentó en el Capítulo 1, esta Tesis ha dividido el proceso constructivo de un entorno virtual destinado a una simulación de conducción en tres fases:

- **Fase 1:** Modelado Conceptual.
- **Fase** 2: Modelado de Datos.
- **Fase** 3: Modelado Físico.

En este capítulo la presente Tesis elabora una solución a la problemática planteada en la segunda fase de creación de un entorno virtual: el Modelado de Datos. En esta fase se afronta el procesado de todas las fuentes de información iniciales para generar las estructuras de datos de partida requeridas para la creación del entorno destinado a una simulación de conducción terrestre.

La generación totalmente automática de dicha información es un objetivo muy ambicioso. Las aplicaciones CAD y SIG se ofrecen como herramienta de ayuda, sin embargo son múltiples los problemas a los que hay que hacer frente en una representación virtual de estas características: la falta de estándares, las incoherencias motivadas por la diversidad de fuentes de información, la escasez de datos y errores asociados a los mismos, su variabilidad en el tiempo, unido a las fuertes restricciones a las que se ven sometidos estos entornos, propician el desarrollo de algoritmos específicos. Esto ha fomentado la creación de aplicaciones particularizadas que aumenten la automatización en el tratamiento de la información geográfica. Esta opción, a pesar de requerir de un mayor esfuerzo de desarrollo inicial, agiliza la construcción del entorno y garantiza su perdurabilidad en el tiempo, ofreciendo una máxima versatilidad y un control total de la representación, ya que nunca debe olvidarse que entre los objetivos de una herramienta CAD o SIG no se encuentra el de un correcto y optimizado aspecto visual.

Por otro lado, la creciente disponibilidad de información geográfica ha impulsado el desarrollo de aplicaciones para la generación automática de mundos virtuales, en especial de ciudades. Tanto empresas destinadas al modelado 3d, como Autodesk, con el software Cityscape de PixelActive [200], como empresas destinadas al SIG, como ESRI con CityEngine [201], como empresas del mundo de los videojuegos, como CRYENGINE [202], han lanzado al mercado potentes herramientas capaces de crear escenarios muy realistas. El problema final encontrado a la hora de emplear cualquiera de ellas siempre es el mismo: adaptar esas herramientas al flujo de creación del entorno de un simulador de conducción terrestre de manera que éste sea eficaz y eficiente.

[200] http://www.pixelactive3d.com/Products/CityScape [Última consulta: 11 Enero 2013]

[201] http://www.esri.com/software/cityengine [Última consulta: 11 Enero 2013]

[202] http://www.crydev.net/dm_eds/download_detail.php?id=4 [Última consulta: 11 Enero 2013]

La bibliografía sobre la metodología elegida para automatizar en la mayor medida posible esta fase es escasa. En líneas generales se puede hablar de dos ramas de actuación. La primera se apoya en el empleo de herramientas comerciales. Este es el caso por ejemplo de los simuladores STISIM [203]. Tomando como herramienta base ArcGis de ESRI y con la ayuda de scripts (ArcObjects [204]) , programan la algorítmica que su simulador requiere. Esta solución trae el inconveniente por un lado de requerir de una licencia, aumentando así los costos y por otro lado, de depender de una herramienta externa.

La segunda línea de actuación busca crear un sistema de información geográfica ad hoc, que se adapte a los requerimientos de un simulador de conducción terrestre. Esta es la solución llevada a cabo por esta Tesis y por entidades dedicadas al desarrollo de simuladores virtuales de conducción como Oktal con su herramienta SCANER™ Studio[205] y NADS, con su Scenario Authoring Tools [206] . También es la iniciativa propuesta en la 25th Conferencia Internacional de Cartografía (ICC 2011) por empresas como Siradel, dedicada al tratamiento de información geográfica, en colaboración con ECA Faros, dedicada a los simuladores de conducción, con su proyecto PlatSim [207].

Siguiendo esta segunda rama de actuación, esta Tesis propone y desarrolla para dar una solución a esta fase, un sistema consistente en varios criterios de organización, corrección, procesamiento y almacenamiento de la información, garantizando siempre su integridad en la representación virtual. El objetivo es crear una metodología modular, que pueda crecer en función de las necesidades del simulador y que permita una manipulación orientada a objetos de la información geográfica. Esto se ha visto plasmado en el desarrollo de una librería en C++ [208], que estructura el entorno en una jerarquía de clases, cuyos objetos, serán almacenados de manera permanente en una base de datos para su posterior consulta y/o edición.

Si bien hasta el momento el proceso constructivo de todo entorno virtual pasaba por la creación inicial del terreno, al que posteriormente se añadían mediante líneas de ruptura los diferentes elementos lineales (carreteras, vías, etc), en el sistema planteado por esta Tesis, las trayectorias circulatorias se convierten en el pilar constructivo. Las trayectorias, elemento clave en toda simulación virtual de conducción terrestre, juegan en la Tecnología Modular un doble papel. Por un lado, como en toda simulación de este tipo, guían el tráfico y por otro lado sirven de líneas directrices para el posicionamiento modular. De esta manera, la solución propuesta por esta Tesis a esta fase sigue los siguientes pasos:

- **Resolución de los problemas geométricos del entorno**: esta Tesis desarrolla para ello un Módulo de Ajuste Geométrico. Dicho módulo procesa la información de partida mediante algoritmos de consolidación, gestión de cortes y suavizado, obteniendo así la definición del entorno en base a una serie de estructuras geométricas base (biarcos, NURBS, nubes de puntos) que respetan las diversas restricciones a las que se ven sometidos este tipo de entornos: restricciones de la normativa circulatoria y de los subsistemas de simulación y motor gráfico así como del proyecto específico que se esté

[203] Haunert, J.H., Brenner, C. & Neidhart, H. Using a geographic information system for the generation of driving Simulator Scenes. Advances in Transportation studies – An International Journal, Special Issue 2005.

[204] Zeiler, M. Exploring ArcObjects, Volume (2)-Applications and Cartography, Redlands, USA. ESRI Press, 2001. ISBN 978-1589480001.

[205] http://www.scanersimulation.com/ . [Última consulta: 11 Enero 2013]

[206] http://www.nads-sc.uiowa.edu/media/pdf/NADS_ISAT.pdf [Última consulta: 11 Enero 2013]

[207] Despine, G., Baillard, C. Realistic Road Modelling for Driving Simulators using GIS Data. Lecture Notes in Geoinformation and Cartography, 2011. Vol 6, n. 7, pp 431-448.

[208] Anexo 6.

desarrollando.

- **Definición de la red topológica del entorno**. Para ello esta Tesis ha creado un Módulo de Generación Topológica cuyo elemento base es el nodo. La diversidad de nodos da lugar a diversos tipos de conexiones, lo que implica definiciones geométricas y funcionalidades específicas. Estos nodos son definidos en base a la normativa circulatoria ferroviaria y vehicular. Por otro lado, este tipo de entornos se ven sometidos a continuas reestructuraciones que es necesario incorporar de manera eficiente, sin motivar inconsistencias con las zonas que no es necesario modificar. Esto solo es posible a través de la manipulación automática de una base de datos que guarde todas las relaciones topológicas existentes entre los diversos elementos del entorno.

- **Definición del entorno circundante**. La extracción y consolidación de los perfiles transversales que componen el entorno así como sus texturas y la definición de la señalización y demás elementos a incluir en la representación virtual permitirán establecer las familias modulares y mallas que se emplearán en la simulación.

Posteriormente, la Algorítmica de Posicionamiento Modular, que será descrita en el Capítulo 5, será la encargada de colocar todos los módulos de entorno definidos en el Capítulo 3, lo que tras la incorporación de elementos puntuales (árboles, edificios, mobiliario urbano..) y la jerarquización de todo el escenario en un grafo de la escena, finalizará el proceso de construcción del entorno.

Una vez procesada toda la información del entorno, ésta es almacenada en bases de datos relacionales y difundida a los diversos subsistemas involucrados en la simulación virtual: simulación, motor gráfico y sector infografista. La difusión de esta información ha motivado la creación de diversos protocolos de comunicación [209].

A continuación se hace un breve repaso del estado del arte llevado a cabo para la resolución de esta fase. Posteriormente se detallará la solución ofrecida por esta Tesis a la misma.

4.2 PROCESAMIENTO DE LA INFORMACIÓN INICIAL REQUERIDA PARA LA GENERACIÓN DEL ENTORNO VIRTUAL.

Las etapas que habitualmente se siguen en la creación de un entorno virtual son las siguientes:

- Extracción de datos y generación del terreno.
- Extracción de datos e inserción de elementos lineales.
- Extracción de datos, modelado y posicionamiento de elementos anejos.
- Adición de las características superficiales mediante el texturado.

La Figura 4.1 muestra un esquema del proceso constructivo de dichos entornos.

[209] Anexo 2.

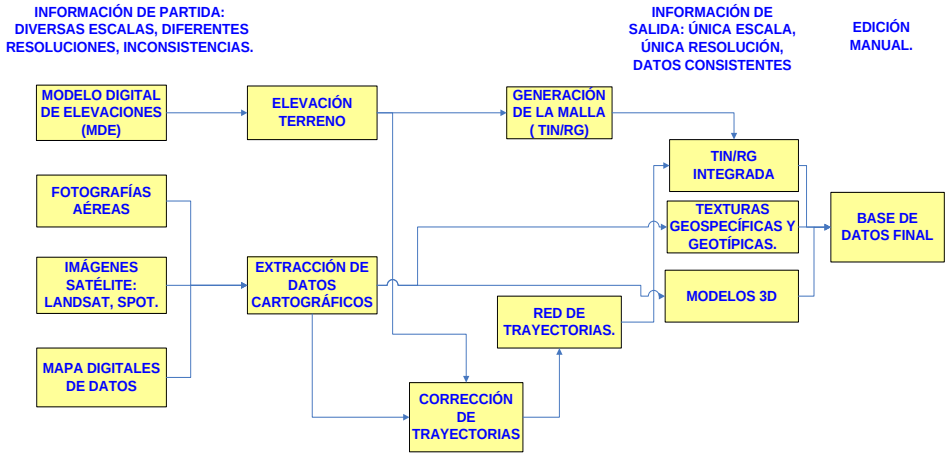


Figura 4.1. Proceso constructivo de un entorno virtual.

A continuación se realiza un repaso del estado del arte de cada una de las etapas del proceso constructivo de un entorno virtual, distinguiendo los casos de generación de entornos reales y ficticios e indicando las herramientas existentes en el mercado que pueden contribuir a su desarrollo en función de las características de la representación.

4.2.1 EXTRACCIÓN DE DATOS Y GENERACIÓN DEL TERRENO.

El proceso de generación del terreno, tanto si éste es real como ficticio, sigue un gran paralelismo con el de su optimización gráfica. De hecho los términos SIG y videojuego cada vez aparecen más unidos en la literatura [210], por utilizar tecnologías muy similares aunque con objetivos distintos. En ambos casos se trata de buscar estructuras de datos y algoritmos que manejen eficazmente un gran volumen de información, en un caso para agilizar los procesos de búsqueda y procesamiento analítico de datos y en otro para garantizar una adecuada velocidad de renderizado e interacción con el usuario.

En el caso de generación de entornos reales, la elevación del terreno se obtiene a partir de un modelo digital de elevaciones (MDE). Un MDE puede definirse como una estructura numérica de datos que representa la distribución espacial de la altitud de la superficie terrestre para un determinado dominio D [211]. Matemáticamente se puede representar como una función bivariable continua z=f(x,y), donde z representa la altitud del terreno en el punto de coordenadas (x,y) y f es una función que relaciona la variable con su localización particularizada para dicho dominio, D: $MDE = (D, f)$.

Sin embargo, en la práctica, la función f no es continua sino que se resuelve a intervalos discretos. Como consecuencia el MDE está compuesto por un conjunto finito y explícito de elementos. Los valores de x e y suelen corresponder con las abscisas y ordenadas de un sistema de coordenadas plano, generalmente un sistema de proyección cartográfica.

[210] Shepherd, Ifan D. H. and Bleasdale-Shepherd, Iestyn D. Chapter 22: Videogames: the new GIS? In: Lin, Hui and Batty, Michael, (ed.) Virtual geographic environments. Science Press, Beijing. 2009. ISBN 978-7030234674.

[211] Felicísimo, A. Modelos Digitales del Terreno. Introducción y aplicaciones en las ciencias ambientales. Pentalfa Eds., Oviedo 1994. ISBN 978-8478484751.

Esta discretización trae como consecuencia una pérdida de información que incrementa el error del MDE. Con el fin de minimizar este error y otros posibles efectos secundarios indeseables de esta discretización, como la dificultad de manejo de la información y el elevado tamaño de los archivos resultantes, se han investigado numerosos formatos para el almacenamiento y representación de la altitud del terreno.

Las posibles estructuras de datos en los MDE se pueden clasificar en dos modelos:

- **Vectorial:** basado en entidades u objetos geométricos definidos por las coordenadas de sus nodos y vértices. Dentro de este grupo las estructuras a destacar son:
  - **Contornos**: se trata de polilíneas de altitud constante. Las dificultades en su manejo informático las convierten en una mera fuente de información que posteriormente será compilada en otros modelos, generalmente una TIN o una malla regular.
  - **TIN**: red irregular de triángulos adosados.
- **Raster:** basado en localizaciones espaciales a las que se asigna un valor de la variable para la unidad elemental de superficie. Dentro de este grupo destacan:
  - **Matrices regulares**: malla de celda cuadrada.
  - **Quadtrees**: estructura jerárquica matricial.

Antes de optar por una determinada estructura de datos, es necesario tener en cuenta sus fuertes implicaciones. El elegir una determinada estructura de datos implica tener que diseñar algoritmos específicos para esta estructura, de manera que quede garantizada una óptima gestión informática. Por otro lado, estas estructuras van a determinar la información a representar y la información a descartar y como consecuencia el mayor o menor volumen de los archivos que la almacenan. Finalmente, el formato elegido supone aceptar las limitaciones de las aplicaciones informáticas que vayan a gestionar la información. La Figura 4.2 muestra una comparativa realizada por Bolstad [212] entre las estructuras de datos raster y vectorial.

Characteristic	Raster	Vector
data structure	usually simple	usually complex
storage requirements	large for most data sets without compression	small for most data sets
coordinate conversion	may be slow due to data volumes, and may require resampling	simple
analysis	easy for continuous data, simple for many layer combinations	preferred for network analyses, many other spatial operations more complex
positional precision	floor set by cell size	limited only by quality of positional measurements
accessibility	easy to modify or program, due to simple data structure	often complex
display and output	good for images, but discrete features may show "stairstep" edges	map-like, with continuous curves, poor for images

Figura 4.2. Comparativa entre modelos vectoriales y raster.

Las estructuras de datos más empleadas hasta el momento en los sistemas de información geográfica son las mallas regulares (*Regular Network, RG*, Figura 4.3) y las mallas irregulares de

[212] Bolstad, P.GIS fundamentals : A First Text on Geographic Information Systems. Second Edition. Elder Press, Minneapolis-St. Paul, 2005. ISBN: 978-0971764736.

triángulos (*Triangulated Irregular Network, TIN*, Figura 4.4.). Las ventajas e inconvenientes de cada uno de estos tipos de mallas es un asunto muy discutido [213]. Hasta hace unos años el formato TIN era considerado el idóneo. Su variable resolución en función de la sinuosidad del relieve permitía por un lado optimizar el número de vértices de la malla y por otro lado una precisa introducción de elementos característicos del entorno, como pueden ser carreteras, ríos,..etc. Este sigue siendo el formato preferido por las aplicaciones comerciales dedicadas a la generación de terrenos [214]. Sin embargo actualmente la gran resolución alcanzable en los actuales MED, permite la generación de formatos híbridos (Figura 4.5) que integren dichos elementos[215], ofreciendo a su vez la ventaja de una cómoda manipulación, debido a la sencillez de esta estructura de datos.

La cantidad de datos vectoriales que sea necesario añadir al terreno será un factor decisivo a la hora de elegir la estructura más idónea, siendo la estructura TIN la más empleada a medida que el número de datos vectoriales aumenta.

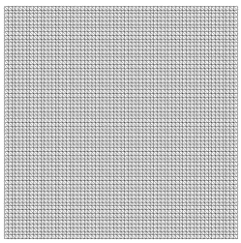


Figura 4.3. Malla Regular (RG).

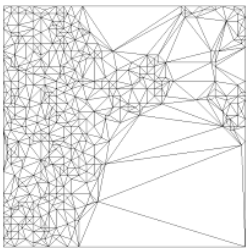


Figura 4.4. Malla Irregular (TIN).

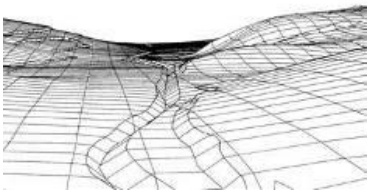


Figura 4.5. Formato Híbrido.

En el caso de generación de terrenos ficticios, los procedimientos existentes son muy numerosos [216]. Entre los primeros algoritmos desarrollados en este ámbito se encuentran los basados en la subdivisión recursiva [217] y en la composición de funciones [218] . Otros, como los basados en técnicas fractales [219] y los algoritmos que pretenden simular el efecto provocado por fenómenos físicos [220] continúan con una activa labor de investigación. Recientemente destaca la

[213] Thurston, J. Looking back and ahead: The Triangulated irregular network (TIN). GEOinformatics, 7. 2003. pp. 32-35.

[214] http://www.presagis.com/products_services/products/ms/content_creation/terra_vista/# [Última consulta: 11 Enero 2013]

[215] Pasado y futuro de los modelos digitales del terreno: mallas regulares y formato híbrido. Mapping interactivo. Revista Internacional de Ciencias de la Tierra. 2005. ISSN 1131-9100, n. 101, pp. 25-29.

[216] Smelik, R. M., De Kraker, K. J. and Groenewegen, S. A. A Survey of Procedural Methods for Terrain Modelling. Proceedings of the CASA Workshop on 3D Advanced Media in Gaming and Simulation (3AMIGAS), Amsterdam, The Netherlands. 2009.

[217] Miller, Gain S.P. The definition and rendering of terrain maps. In Proceedings of the 13st Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1986. Vol. 20, pp 39-48

[218] Perlin, K. An Image Synthesizer. In Proceedings of the 12st Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1985. Vol. 19, pp 287-296.

[219] Belhadj, F. Terrain Modeling: a Constrained Fractal Model. In Spencer, S. N., editor, AFRIGRAPH '07: Proceedings of the 5th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa, 2007. pp 197-204.

[220] St'ava, O., Benes, B., Brisbin, M., and Krivanek, J. Interactive terrain modeling using hydraulic erosion. In Proceedings of Symposium on Computer Animation, 2008, pp 201-210.

aportación de De Carpentier y Bidarra[221], que haciendo uso de la GPU proponen una metodología de generación de terrenos mediante la aplicación de herramientas pincel (*brushes)*. Y la aportación de Peytavie et al[222], que permiten la generación de terrenos complejos que incluyen túneles, cuevas..etc. Pero a pesar de que la generación procedural de terrenos ha sido un área de investigación muy activa en los últimos treinta años, el usuario rara vez tiene un control total sobre el aspecto final del terreno, ya que los parámetros a controlar suelen ser numerosos y poco intuitivos. Por otro lado, la generación completa de un terreno con incorporación de elementos lineales siempre suele requerir de un retoque manual.

Como se comentó en el apartado 2.4.3 del Capítulo 2, las herramientas comerciales (COTS, Commercial off the Shelf) dedicadas a la generación de terrenos son muy numerosas [223], sin embargo, desde el punto de vista de las simulaciones de conducción terrestre, solo una pequeña fracción de ellas resultan eficientes. Esto es debido a las múltiples dificultades que acampañan a entornos de este tipo: introducción de una enorme red de trayectorias con fuertes restricciones geométricas y topológicas; edición de las mismas garantizando en todo momento la coherencia de la red circulatoria; almacenamiento permanente de toda la información relacionada con la malla de terreno; optimización para permitir una simulación en tiempo real. Como consecuencia de todas estas exigencias únicamente han tenido éxito en este ámbito las herramientas de generación de terrenos que han sido incorporadas al proceso constructivo del entorno virtual, constituyendo un paso intermedio del mismo. Este es el caso de las soluciones aportadas por Presagis, con Terra Vista [224], por TerraSim, con TerraTools [225] o por Planetside, con Terragen [226]. Estas empresas han desarrollado una gama de productos que abarcan todo el proceso constructivo del entorno virtual, estandarizando los formatos de entrada y salida de cada etapa, lo que las ha convertido en las herramientas preferidas de los desarrolladores que optan por apoyarse en aplicaciones existentes.

4.2.2 EXTRACCIÓN DE DATOS E INSERCIÓN DE ELEMENTOS LINEALES.

La extracción y compatibilización de la información relacionada con los elementos lineales del entorno es la tarea más compleja y que requiere de una mayor inversión de tiempo [227] [228] [229] [230] . En el caso de las simulaciones de conducción terrestre, la extracción automática de

[221] De Carpentier, G. and Bidarra, R. Interactive GPU-based Procedural Height field Brushes. In Proceedings of the 4th International Conference on the Foundation of Digital Games, Florida, USA. 2009.

[222] Peytavie, A., Galin, E., Merillou, S., and Grosjean, J. Arches: a Framework for Modeling Complex Terrains. In Eurographics Proceedings. Eurographics Association 2009.

[223] http://www.vterrain.org/Packages/Com/index.html [Última consulta: 11 Enero 2013]

[224] http://www.presagis.com/products_services/products/ms/content_creation/terra_vista# . [Última consulta: 11 Enero 2013]

[225] http://www.terrasim.com/products/ . [Última consulta: 11 Enero 2013]

[226] http://www.planetside.co.uk/ [Última consulta: 11 Enero 2013]

[227] Rajani Mangala, T., G Bhirud, S. A New Automatic Road Extraction Technique using Gradient Operation and Skeletal Ray Formation. International Journal of Computer Applications, 2011. Vol. 29, n. 1, pp. 17-25.

[228] Gurumurthy, R., Omkar, S.N., Senthilnath, J. and Reddy, P.. Automatic Extraction of Road Networks based on Normalized cuts and Mean Shift method for high resolution Satellite Imagery. International Journal of Advanced Engineering Sciences and Technologies, 2011. Vol. 3, n. 2, pp. 115 – 121.

carreteras (Figura 4.6) encuentra su mayor hándicap en la correcta interpretación de las conexiones [231]. Estas definirán la red topológica que permitirá la circulación a su través.

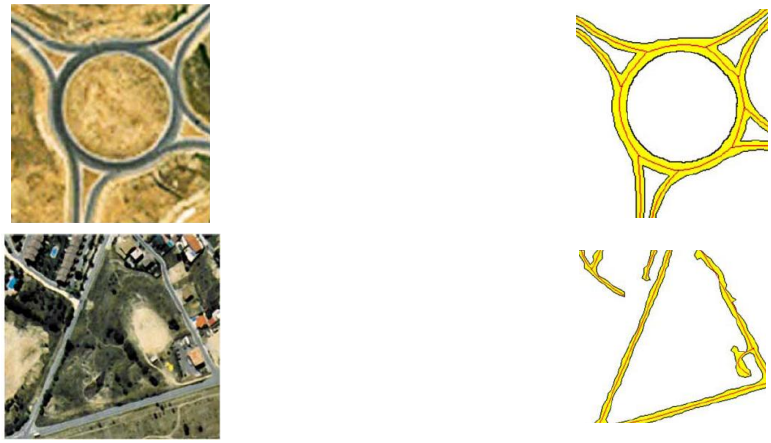


Figura 4.6. Ejemplos de resultados de extracción automática presentados por J.Mena [232].

Las fuentes de información destinadas a este fin son varias. Por una lado se encuentran las imágenes adquiridas por técnicas de percepción remota, como pueden ser fotografías aéreas e imágenes satelitales (Landsat, SPOT). Previo a su empleo, estas imágenes necesitan ser tratadas con diversas técnicas que suplan las carencias y corrijan las distorsiones en ellas existentes como consecuencia de las características inherentes a su toma de datos [233].

Cuando la resolución de estas fuentes es insuficiente, se hace necesaria su compaginación con otras más precisas, como son los mapas digitales de datos [234]. Sin embargo, a pesar de su mayor resolución, las imprecisiones asociadas al proceso de digitalización pueden seguir ocasionando conflictos (hay que tener en cuenta que el objetivo para el cuál estos planos fueron creados, no es en la mayoría de casos, la extracción de rutas con la precisión exigida por una representación virtual). En este caso, es necesario recurrir a la introducción explícita de la geometría conflictiva, siendo la compatibilización de todas las fuentes el punto más delicado.

Una vez solventados todos los errores e incompatibilidades asociados a la extracción de elementos lineales y disponible así toda su información, es necesaria su integración en la malla tridimensional del terreno. Esto requerirá resolver nuevas incongruencias, partes de ellas generadas por la diferente precisión del modelo digital de elevaciones (MDE) y de los mapas digitales de datos.

Existen tres técnicas diferentes a seguir a la hora de integrar elementos lineales, como ríos o carreteras, en la malla representativa del terreno:

---

[229] Li, Y. and Briggs, R. Automatic Extraction of Roads from High Resolution Aerial and Satellite Images with Heavy Noise. World Academy of Science, Engineering and Technology, 2009. Vol. 54, pp.416-422.

[230] Mena, J. B. State of the art on automatic road extraction for GIS update: a novel classification, Pattern Recognition Letters 2003. Vol. .24, n.16, pp.3037-3058.

[231] Koutaki, G., Hu, Z., Uchimura, K. Automatic Road Extraction Based on Intersection Detection in Subsurban Areas. Journal of Imaging Science and Technology 2005. Vol.49, n.2, pp.163-169.

[232] Mena, J. B., Malpica, J.A. An automatic method for road extraction in rural and semiurban areas starting from high resolution satellite imagery. Pattern Recognition Letters 2005. Vol. 26, n. 9, pp. 1201 – 1220.

[233] Hinz, S. Automated road extraction in urban scene and beyond. International Archives of Photogrammetry and Remote Sensing, 2004. Vol. 35, n. B3, pp. 349-355.

[234] http://www.vterrain.org/Culture/vector.html [Última consulta: 11 Enero 2013].

- Los elementos lineales forman parte de la textura del terreno [235] [236] [237]. Esta metodología se puede emplear en caso de que los elementos a representar no requieran de un exhaustivo nivel de detalle. Este es el caso de los escenarios generados para el proyecto NatSim (Nature Simulation) [238], como muestra la siguiente figura.

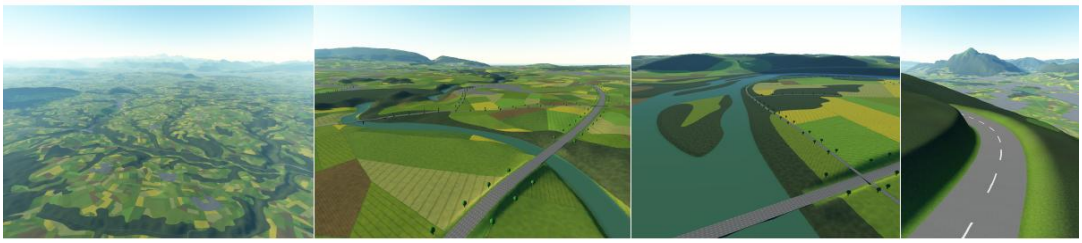


Figura 4.7. Escenario generado para el proyecto NatSim.

- Siguiendo el terreno se coloca sobre el mismo una geometría [239] [240] [241], ofreciendo las ventajas de ser un proceso simple y el inconveniente de producirse efectos de z-buffer o parpadeo indeseados. Al igual que en el caso anterior esta metodología se puede emplear en los casos en los que los elementos a insertar se perciban con un bajo nivel de detalle [242] , como es el caso de los simuladores de vuelo (Figura 4.8).

Figura 4.8. Ejemplo de representación de elementos lineales en un simulador de vuelo.

[235] Bruneton, E., Neyret, F. Real-Time Rendering and Editing of Vector-based Terrains. Computer Graphics Forum 2008. Pp 311-320.

[236] Kersting, O., and Dollner, J. Interactive 3D Visualization of Vector Data in GIS. In Proceedings Of 10th ACM Int. Sym. on Advances in Geographic Information Systems. 2002.pp. 107-112.

[237] Döllner, J., Baumann, K., and Hinrichs, K. Texturing techniques for terrain visualization. In VISUALIZATION '00: Proceedings of the 11th IEEE Visualization 2000 Conference (VIS 2000).

[238] http://www.irit.fr/NatSim [Última consulta: 11 Enero 2013].

[239] Agrawal, A., Radhakrishna, M. and Joshi, R. Geometry-based mapping and rendering of vector data over LOD phototextured 3D terrain models. In Proceedings of WSCG, 2006.pp 787–804.

[240] Schneider, M., Guthe, M., and Klein, R. Realtime rendering of complex vector data on 3d terrain models. In Thwaites, H., editor, The 11th International Conference on Virtual Systems and Multimedia (VSMM2005), pp 573–582.

[241] Wartell, Z., Kang, E., Wasilewski, T., Ribarsky, W. and Faust, N. Rendering vector data over global, multi-resolution 3d terrain. In VISSYM '03: Proceedings of the symposium on Data visualization 2003. pp 213–222.

[242] Szofran, A. Global Terrain Technology for Flight Simulation. Microsoft ACES Game Studio.Game Developers Conference 2006. San Jose, California.

- Mediante el empleo del algoritmo *stencil shadow volume* [243]. Este procedimiento sigue tres pasos: generación de un poliedro a partir del vector de datos; creación en el *stencil buffer* de una máscara a partir de dicho poliedro; aplicación de la máscara a la escena, para renderizar el vector de datos. Este método, empleado también en la visualización terrenos que no requieran de gran realismo en la representación de las trayectorias, presenta la ventaja de ser independiente del algoritmo de renderizado del terreno, pero a costa de un mayor costo computacional.

Figura 4.9. Visualización de carreteras mediante aplicación del algoritmo stencil shadow volume [244].

- La geometría del elemento aparece fusionadas al terreno, con la ventaja de que en este caso los límites terreno-elemento aparecen perfectos y sin efectos Z-buffer y con el inconveniente de requerir una fase de preprocesamiento larga y tediosa: compatibilización de alturas, interpolación de éstas en los puntos necesarios como consecuencia de la diferente resolución entre el modelos digital de elevaciones y los datos de la obra lineal y retriangulación de la malla en la zona adyacente a la inserción de la obra lineal para adaptarse a la mayor resolución de ésta. Esta es la única opción que ofrece la calidad visual exigida a una simulación de conducción terrestre.

La introducción de elementos lineales ha sido siempre el punto débil de la mayoría de las aplicaciones comerciales. Esto es debido por un lado al alto grado de precisión y suavizado exigido por las simulaciones de conducción terrestre. Por otro lado, la necesidad de un almacenamiento estructurado tanto de la información geométrica como topológica de la red de trayectorias con el fin de poder manipularla y transmitirla al resto de subsistemas involucrados en la simulación (motor gráfico, módulo de conducción), hace que el empleo de aplicaciones comerciales siempre tenga que verse completado posteriormente con un gran trabajo manual.

La solución a este problema consiste en la creación de herramientas específicas que automaticen al máximo esta tarea de creación de la red de trayectorias. Esta es la medida llevada a cabo por esta Tesis y por entidades como Oktal con su herramienta SCANER™ Studio Scenario [245] y NADS, con su Interactive Scenario Authoring Tool [246]. De la misma manera, para el caso de generación de redes de trayectorias ficticias, diversas empresas dedicadas a la creación de simuladores de conducción han desarrollado sus propias herramientas como CORYS, que ha

[243] Vaaraniemi, M., Treib, M., Westermann, R. High-Quality Cartographic Roads on High-Resolution DEMs. Journal of WSCG 2011. Pp 41-48.

[244] Schneider, M. and Klein, R. Efficient and accurate rendering of vector data on virtual landscapes. Journal of WSCG, 2007. Vol 15, pp 1-3.

[245] http://www.scanersimulation.com/ . [Última consulta: 11 Enero 2013]

[246] http://www.nads-sc.uiowa.edu/media/pdf/NADS_ISAT.pdf [Última consulta: 11 Enero 2013]

desarrollado el Track Builder Tool [247] y WIVW (Wuerzburg Institute for Traffic Sciences), que ha creado el software SILAB [248].

En cuanto a la generación de escenarios ficticios en este ámbito, hay que destacar las numerosas investigaciones que en los últimos años están teniendo lugar en la generación procedural de trayectorias, en particular de carreteras [249] [250]. Aunque todavía se trata de métodos poco intuitivos, con muchos parámetros a controlar y que suelen requerir de un retoque manual final.

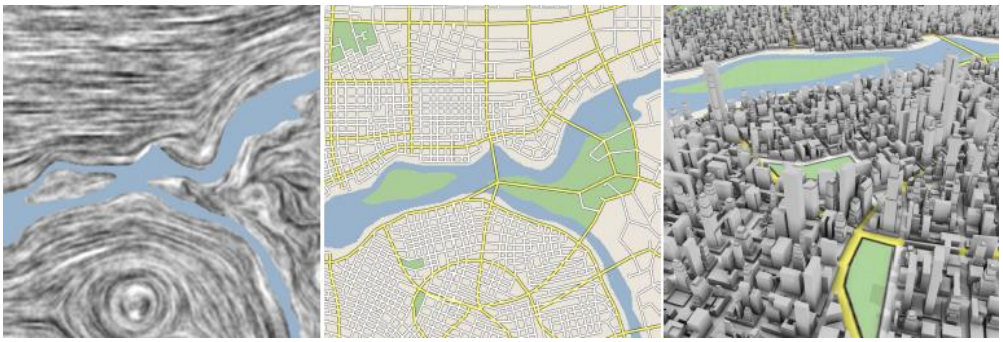


Figura 4.10. Generación procedural de calles [251].

4.2.3 EXTRACCIÓN DE DATOS, MODELADO Y POSICIONAMIENTO DE ELEMENTOS ANEJOS.

Las variadas y complejas geometrías de los restantes elementos constituyentes del entorno, han impedido la creación de un formato estandarizado para su descripción, lo que dificulta su extracción y modelado automático [252]. En particular, la extracción automática de edificios ha sido y es, motivo de gran investigación [253] [254] [255] [256]. Su definición requiere de un

[247] http://www.corys.com/The-Tools---443.html [Última consulta: 11 Enero 2013]

[248] http://www.wivw.de/ProdukteDienstleistungen/SILAB/ [Última consulta: 11 Enero 2013]

[249] Lipp, M., Scherzer, D., Wonka, P., Wimmer, M. Interactive Modeling of City Layouts using Layers of Procedural Content. Computer Graphics Forum 2011 (Proceedings of Eurographics 2011). Vol. 30. n.2, pp.345–354.

[250] Galin, E., Peytavie, A., Guérin, E., Benes, B. Authoring Hierarchical Road Networks. Computer Graphics Forum 2011.(Proceedings of Eurographics 2011). Vol. 30, .n.7, pp.2021–2030.

[251] Chen, G., Esch, G., Wonka, P., Müller, P.and Zhang, E. Interactive Procedural Street Modeling. Proceedings of ACM SIGGRAPH 2008 / ACM Transactions on Graphics (TOG), ACM Press, Vol. 29, n.3, 9 pp. 1-10.

[252] Mayer, H. Object extraction in photogrammetric computer vision. ISPRS Journal of Photogrammetry and Remote Sensing 2008. Vol 63, n. 2, pp 213–222.

[253] Kelly, T., Wonka, P. Interactive Architectural Modeling with Procedural Extrusions. In Proceedings of ACM Transactions on Graphics, 2011. Vol. 30, n. 2. pp 1-15.

[254] Xiao, J., Fang, T., Tan, P., Zhao, P., Ofek, E., Quan, L. Image-based façade modelling. In Proceedings of ACM Transactions on Graphics 2008. Vol. 27, n. 5.

[255] Brenner, C. Building reconstruction from images and laser scanning. International Journal of Applied Earth Observations and Geoinformation, 2005. Vol. 6, n. 3-4, pp. 187-198.

completo análisis tridimensional, lo que supone la combinación de imágenes aéreas y terrestres. En [257] [258] se hace una comparativa sobre los diferentes grados de precisión obtenidos en la representación de edificios en función de las fuentes de información empleadas.

En cuanto al modelado de dichos elementos, existen diversas alternativas que permiten toda una gama de grados de automatización y precisión, en función del tipo de representación virtual a la que vayan destinados [259]. La Figura 4.11 muestra un ejemplo del proceso constructivo de un edificio, donde la generación automática obtenida a partir de la combinación de imágenes aéreas y terrestres ha exigido finalmente un retoque manual.

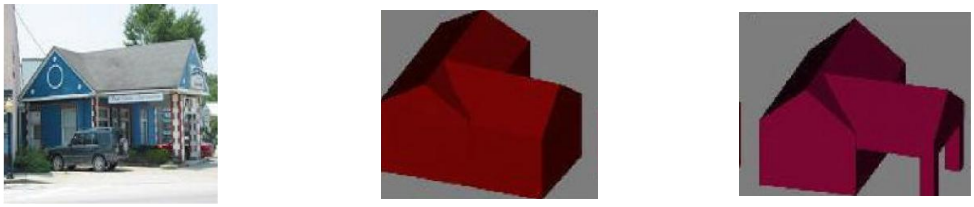


Figura 4.11. Ejemplo de reconstrucción de edificio: modelo real, modelo obtenido por extracción automática y modelo tras la edición manual.

El empleo de técnicas fotogramétricas es el procedimiento más riguroso y costoso, tanto computacional como económicamente [260] [261] (Figura 4.12).

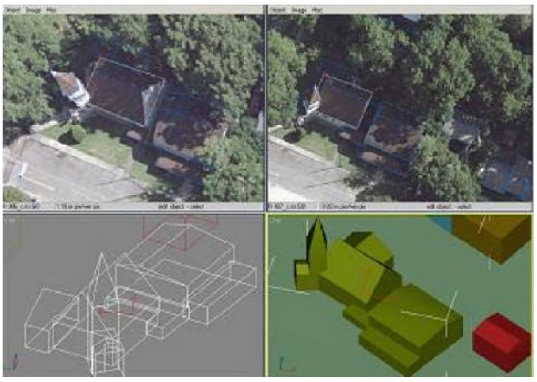


Figura 4.12. Ejemplos de extracción de información y modelado de edificios.

[256] Brenner, C. Towards fully automatic generation of city models. International Archives of Photogrammetry and Remote Sensing, 2000. Vol. 33, n. B3/1, pp. 85-92.

[257] Kaartinen, H., Hyyppä, J., Gülch, E., Vosselman, G., et al. Accuracy of 3D city models: EuroSDR comparison. ISPRS WG III/3, III/4, V/3 Workshop, Laser scanning 2005.

[258] Maillet, G. Flamanc, D. Comparison of Aerial Images, Satellite images and laser scanning DSM in a 3D City Models Production Framework Geo-Imagery Bridging Continents.20th ISPRS Congress, 2004 Istanbul, Turkey .

[259] Hu J. You , S., Neumann, U. Approaches to large-scale urban modeling. IEEE Computer Graphics and Applications, 2003. Vol 23, n. 6, pp 62–69.

[260] Shashi, M. and Jain, K..Use of photogrammetry in 3D modelling and visualization of buildings. ARPN Journal of Engineering and Applied Sciences 2007. Vol. 2, n. 2, pp. 37–40.

[261] Emem, O., Batuk, F. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences 2004. Vol 35; n. 4, pp. 431-436.

En el caso de las simulaciones de conducción, la velocidad a la que el usuario circula a través del entorno dificulta su percepción de éste. Por otro lado, como se comentó en el Capítulo 3, el valor funcional de estos objetos de cara a la simulación es bajo. Esto permite minimizar los recursos gráficos empleados en este ámbito, sustituyendo en la mayor medida posible la complejidad geométrica por imágenes.

El posicionamiento de los elementos anejos, exigirá una previa verificación de la correcta pendiente del terreno en el lugar de su colocación y posteriormente la interpolación de su cota de elevación. En el caso de que la pendiente no sea la idónea, la malla habrá de ser retocada, prestando especial atención a la posible afectación de elementos situados en las inmediaciones. Las restricciones topológicas de la malla elegida, sus tolerancias de deformación admisibles o el número máximo de triángulos permitidos, son factores que contribuirán a la determinación del tipo de modificaciones más idóneas [262].

4.2.4 ADICIÓN DE LAS CARACTERÍSTICAS SUPERFICIALES MEDIANTE EL TEXTURADO.

Finalmente las características superficiales se añadirán a través de la colocación de texturas. Se distinguen dos tipo de texturas: geospecíficas y geotípicas. Las grandes diferencias existentes en el tratamiento de ambos tipos de texturas ha llevado a la distinción entre dos técnicas de mapeado: el tradicional y el basado en la transformación de perspectiva de la imagen (*Image Perspective Transformation*, IPT)[263].

Las texturas geospecíficas, obtenidas a partir de técnicas de percepción remota, dotan al entorno de un aspecto realista. Son imágenes únicas para cada objeto del escenario. Generalmente representan vistas oblicuas de la superficie, por lo que han de ser en primer lugar tratadas para eliminar las distorsiones y posteriormente proyectadas sobre éstas, mediante el empleo de técnicas fotogramétricas. El modelado de los objetos 3D del terreno y estructuras urbanas sobre las que se van a aplicar dichas texturas, se realiza mediante la aplicación de técnicas fotogramétricas. Los elevados requerimientos de memoria de estas texturas, hacen que su uso sea únicamente justificable en el caso de aplicaciones virtuales que requieran un elevado grado de reconocimiento del terreno.

Las texturas geotípicas (Figura 4.13) por el contrario aportan, en distancias cortas, el realismo que las geoespecíficas no son capaces de proporcionar. Para ello definen diferentes prototipos de terreno (tierra, hierba..) generados bien a partir de fotografías terrestres, bien sintéticamente a través de programas de retoque de la imagen o a través de programas específicos para la generación de texturas. Se trata de imágenes de pequeño tamaño que se repiten a lo largo de diversos objetos y superficies. Representan vistas frontales de la superficie, por lo que se pueden aplicar ortogonalmente sobre éstas sin mostrar ninguna distorsión. Los objetos 3D sobre las que se aplican, son modelados con independencia de la textura que se vaya a aplicar.

[262] Luebke, R., Varshney, C., Huebner, W. Chapter 2: Mesh Simplification. Level of Detail for 3d Graphics. Morgan Kaufmann Publishers. 2003. ISBN: 978-1558608382.

[263] Weinhaus F. M. , Devarajan, V. Texture mapping 3d models of real-world scenes, ACM Computing Surveys, 1997. Vol. 29, n. 4, pp. 325-365.

Figura 4.13. Ejemplos de texturas geotípicas.

4.3 EL MÓDULO DE GENERACIÓN TOPOLÓGICA.

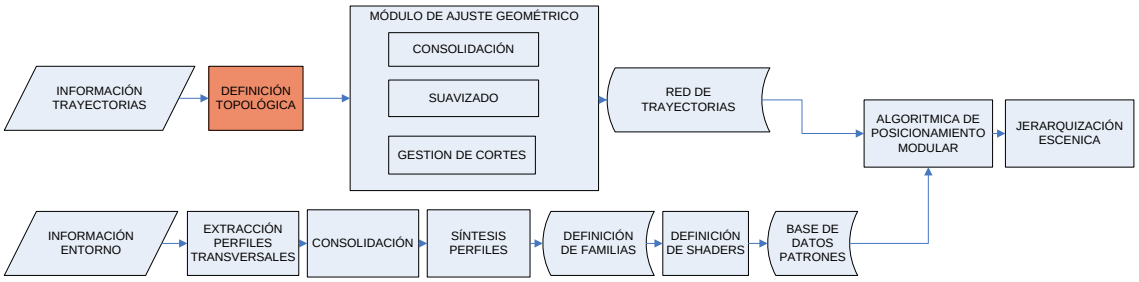


Figura 4.14. Esquema de funcionamiento de la Tecnología Modular.

Buscando la máxima modularidad, versatilidad y escalabilidad posible, esta Tesis ha diseñado un Módulo de Generación Topológica en el que el objeto *nodo* constituye el pilar de su construcción. Se define el *nodo* como el elemento de conexión entre dos trazados. Este *nodo* llevará intrínsecamente asociada una definición geométrica y funcional de manera que permita la generación y circulación a través de cualquier tipo de entorno ferroviario, urbano o interurbano.

Puesto que en este tipo de representaciones virtuales las trayectorias actúan como líneas directrices en la generación del entorno, se puede considerar que estos entornos poseen un carácter predominantemente unidimensional, definiéndose como medida de posicionamiento principal el punto kilómetro o PK.

4.3.1 DEFINICIÓN TOPOLÓGICA DE ENTORNOS FERROVIARIOS.

Los diferentes elementos topológicos que esta Tesis ha creado para definir una red ferroviaria son los siguientes:

- **LDL (Line Description Layout):** objeto que engloba a todos los elementos pertenecientes a un entorno ferroviario. Todos los objetos de un LDL tienen en cuenta al resto de objetos y reaccionan ante sus cambios. Por ejemplo, un cambio en una determinada línea, puede

extenderse sobre entornos, elementos de señalización u otras líneas que hagan referencia a dicha línea.

- **Nodo:** elemento de conexión entre dos trazados. En un entorno ferroviario se han definido los siguientes tipos de nodos (Figura 4.15):
    - **Nodos extremos**: inicial (*TrackBoundaryNode*) y final (*TrackEndNode*). Definen los comienzos y fines de línea que no tienen vinculaciones extremas.
    - **Nodo Simple** (*Simple Node*). Define la conexión de dos trazados.
    - **Nodo Triple** (*Points Node*). Define la conexión de tres trazados.
    - **Nodo Cuádruple** (*Diamond Node*). Define la conexión de cuatro trazados.

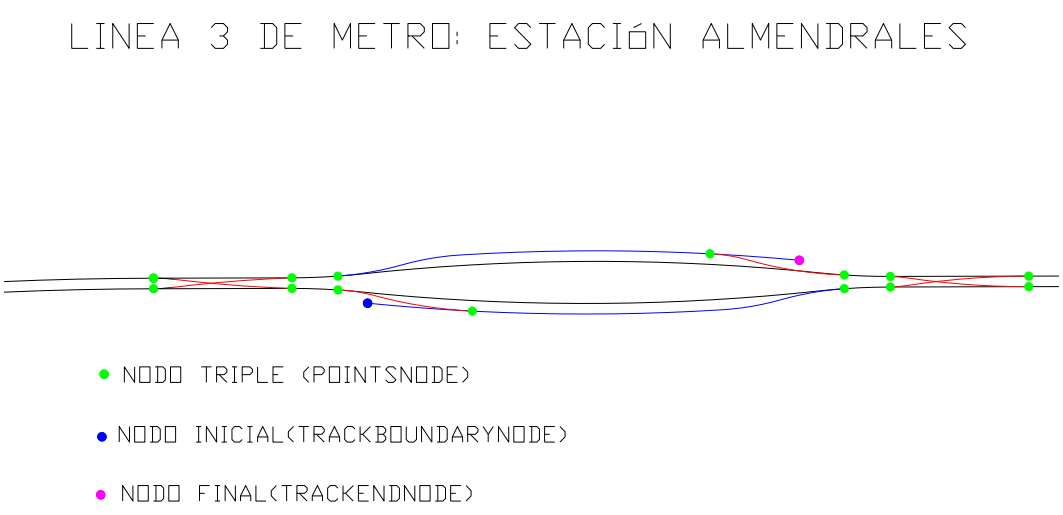


Figura 4.15. Ejemplo de definición topológica ferroviaria: estación de Almendrales.

- **Sección**: tramo circulatorio comprendido entre dos nodos consecutivos.
- **Línea:** lista de secciones. Existirán dos categorías de líneas:
    - **Líneas ficticias**: aquellas curvas imaginarias del escenario que permiten la definición de las restantes líneas mediante relaciones de paralelismo. Tienen una connotación meramente geométrica.
    - **Líneas funcionales**: líneas del escenario a través de las cuáles se puede circular. Estas a su vez, en función de las conectividades de su nodos extremos podrán clasificarse en los siguientes grupos, lo que, como se verá posteriormente, tendrá una serie de repercusiones geométricas:
        - **Líneas normales**: aquéllas cuyos nodos extremos no guardan ninguna vinculación con otras líneas.
        - **Líneas que comienzan en líneas** (*Line begins in Line, LBL*): aquéllas que comienzan con un nodo triple (*PointsNode*).
        - **Líneas que terminan en líneas** (*Line ends in Line, LEL*): aquéllas que terminan con un nodo triple (*PointsNode*).
        - **Líneas que comienzan y terminan en líneas** (*LB&EL*): aquéllas que comienzan y terminan con un nodo triple (*PointsNode)*.

- **Aguja:** tramo circulatorio comprendido entre dos nodos triples consecutivos. Como se verá en el apartado correspondiente a la descripción geométrica, también llevará asociada una definición geométrica específica.

- **PK: punto kilométrico,** medida de posicionamiento para cualquier elemento del entorno**.** Existirán tres tipos de pks: absoluto, de línea y de sección.
    - **El pk absoluto** asegurará una colocación correcta de elementos enfrentados en diferentes líneas. Para su definición es necesaria la existencia de al menos una línea ficticia, que actuará como directriz del entorno, definiendo la geometría del trazado principal respecto del cuál se definirán los paralelismos de las restantes líneas.
    - **El pk de línea,** relativo a la línea, garantizará un correcto posicionamiento relativo dentro de ésta.
    - **El pk de sección**, relativo a la sección, garantizará un correcto posicionamiento relativo dentro de ésta.
- **S: espacio recorrido.** Además de por el punto kilométrico, todo elemento perteneciente al entorno podrá venir posicionado por el espacio recorrido. De la misma manera que con el pk, se definirá un espacio absoluto, de línea o de sección. El tipo de elemento a posicionar determinará el tipo de medida más idónea.

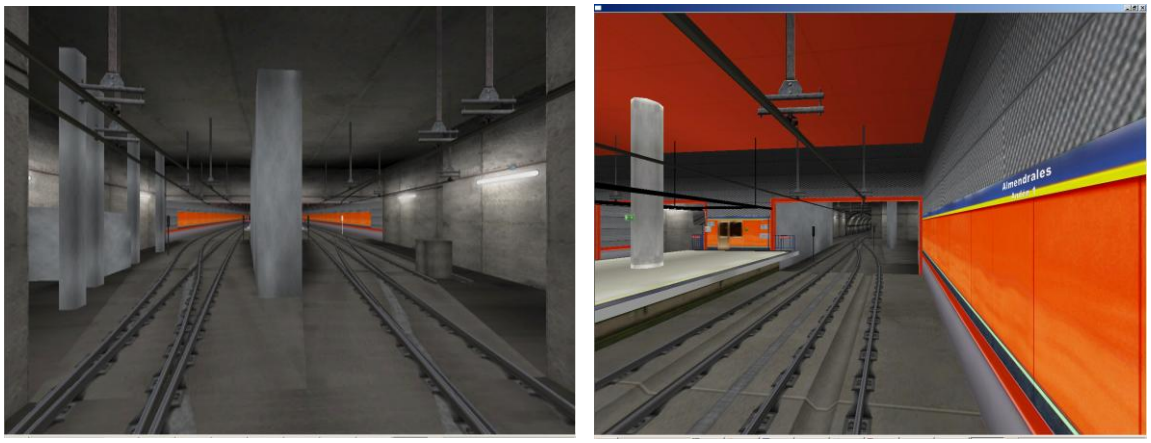


Figura 4.16. Estación virtual de Almendrales: ejemplos de diferentes tipos de líneas ferroviarias.

La definición topológica de un entorno ferroviario comienza con la creación de un objeto LDL al que se irán añadiendo todos los elementos que definen un entorno de estas características. En primer lugar se definirán el conjunto de nodos que definen la red ferroviaria a representar, así como la forma de conectarlos. A partir de ellos se definirán las secciones, líneas y agujas.

4.3.2 DEFINICIÓN TOPOLÓGICA DE ENTORNOS URBANOS.

Los diferentes elementos topológicos que esta Tesis ha creado para definir una red urbana son los siguientes:

- **Network**, objeto que almacena a todos los elementos pertenecientes a un entorno urbano. Al igual que en un LDL, todos los objetos de una network tienen en cuenta el resto de objetos y reaccionan ante sus cambios.

  Nodo: al igual que en un entorno ferroviario constituirá la raíz del proceso constructivo del entorno. Según su funcionalidad se distinguen los siguientes tipos:

  - Nodos cruce: puntos donde confluyen tres o más Curvas Constructivas.

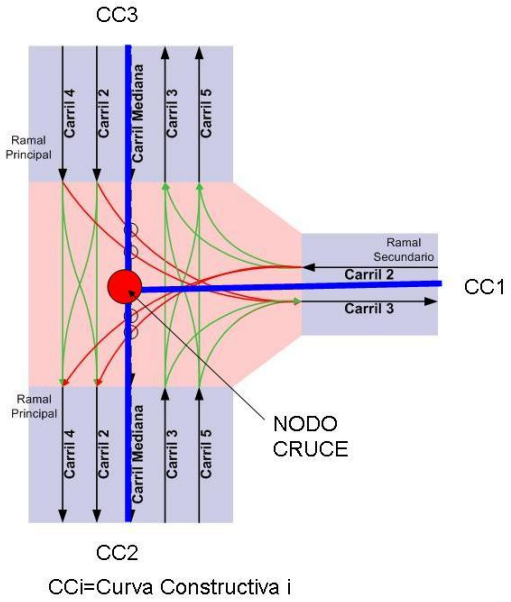


Figura 4.17. Nodo cruce.

  - Nodos de ampliación y/o disminución del número de carriles: puntos donde una Curva Constructiva modifica su número de carriles.

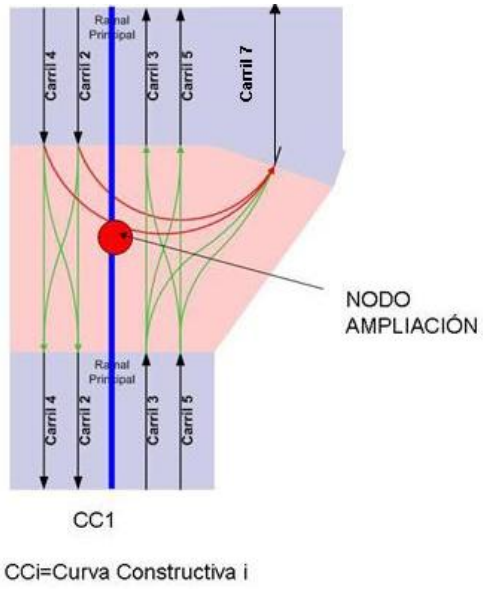


Figura 4.18. Nodo de ampliación.

- Nodos rotonda: puntos donde confluyen dos o más Curvas Constructivas y llevan asociados carriles circulares para transitar de unas a otras.

- Nodos de incorporación y/o salida: puntos donde confluyen dos Curvas Constructivas.

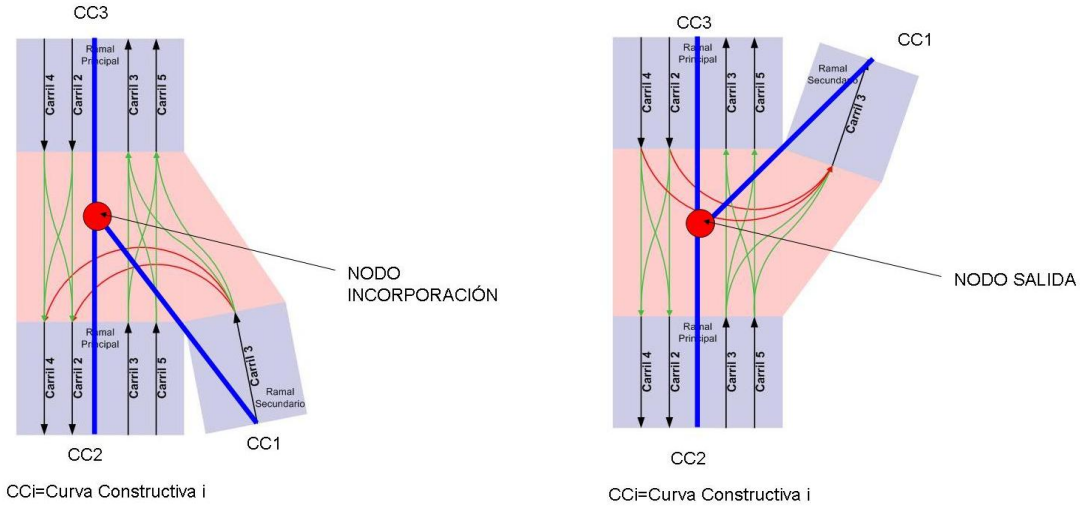


Figura 4.19. Nodos de incorporación y salida.

- **Curva Constructiva**, trazado que une dos nodos y a partir del cuál se definen los carriles ficticios. Su connotación es meramente geométrica. Cada curva constructiva lleva asociada un sentido, que vendrá reflejado en sus conectividades y en su nomenclatura: Ni_Nf, siendo Ni el nombre asociado a su nodo de inicio y Nf al de su nodo fin. Su sentido será por tanto entrante a la intersección definida por el nodo Nf.

- **Ruta**, posible trayectoria a seguir por un vehículo en el interior de una intersección.

- **Carril**, objeto que va a jugar el mismo papel en un entorno urbano que la línea en el ferroviario. Al igual que en un entorno ferroviario se distingue entre:

    - **Carriles ficticios**: serán aquellas curvas imaginarias del escenario que permiten la definición de los restantes carriles mediante relaciones de paralelismo. Tienen una connotación meramente geométrica.

    - **Carriles funcionales**: carriles del escenario a través de los cuáles se puede circular. Los carriles con numeración par, llevan el mismo sentido que su curva constructiva y los carriles impares sentido contrario.

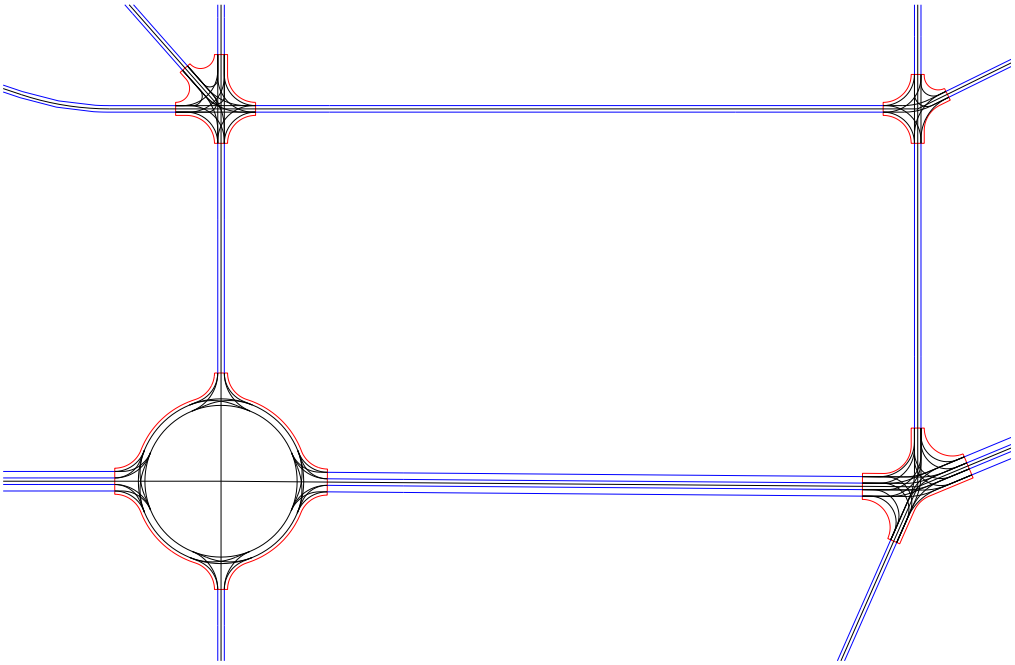


Figura 4.20. Carriles (azul) y Rutas (negro).

- **Ramal**, conjunto de carriles existentes entre dos nodos. El número de carriles en un ramal es siempre constante.
- **Calle**, lista de ramales.
- **Intersección**, área alrededor de un nodo limitada por los ramales que confluyen en él.

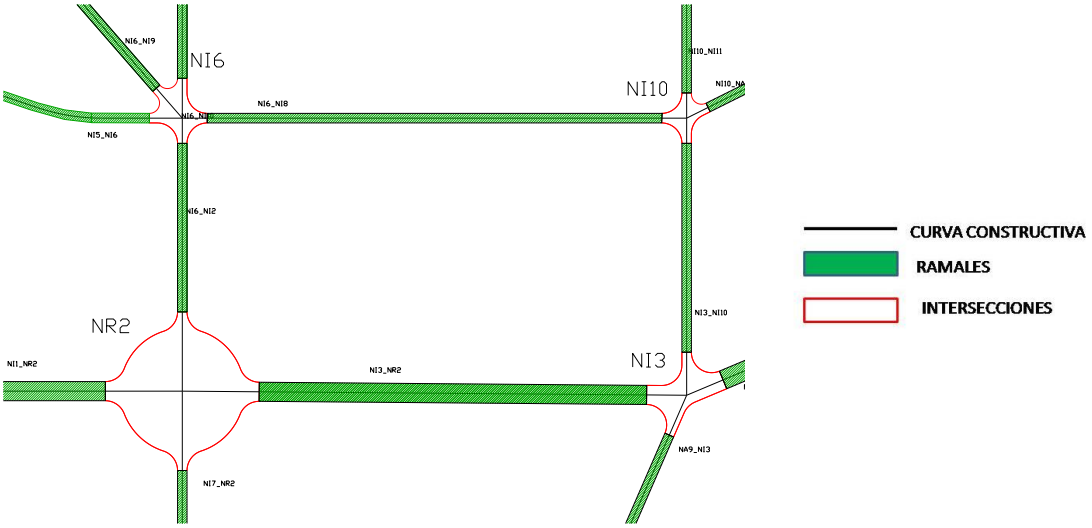


Figura 4.21. Curvas Constructivas, Ramales, Intersecciones.

La definición topológica de un entorno urbano comienza con la creación de un objeto Network al que se irán añadiendo todos los elementos que definen un entorno de estas características.

En primer lugar se definirán el conjunto de nodos que definen la red urbana a representar, así como la forma de conectarlos. A partir de ellos se definirán las curvas constructivas. El tipo de nodo que define una intersección, las curvas constructivas que confluyen en ella y la información de los radios de acuerdo permitirán generar el contorno de la intersección. A partir de éste se delimitarán los ramales que confluyen en la misma, mediante la selección del tramo de curva constructiva comprendida entre los contornos de las intersecciones que la limitan. Definidos los pks de inicio y fin de los ramales, quedarán delimitados los carriles que los integran.

Finalmente se agruparán los ramales según la definición de calles dada y se procederá la cálculo del contorno de las intersecciones, que por su falta de repetibilidad serán normalmente modeladas mediante mallas ad hoc realizadas automáticamente.

4.4 EL MÓDULO DE AJUSTE GEOMÉTRICO.

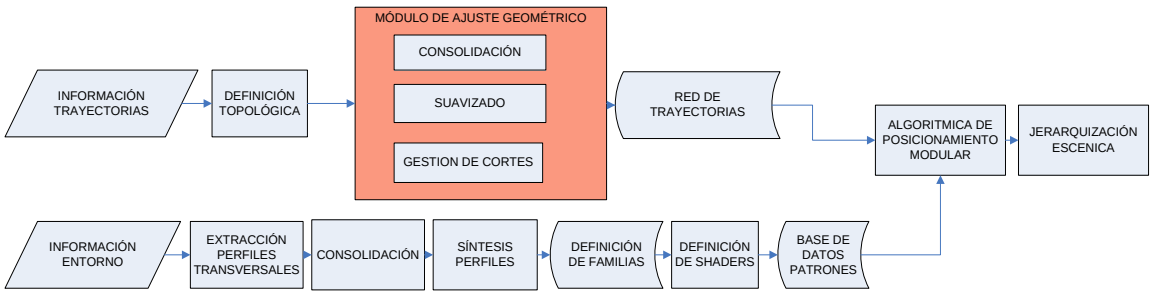


Figura 4.22. Esquema de funcionamiento de la Tecnología Modular.

Como ya se comentó en la introducción de este capítulo, las trayectorias circulatorias constituyen el pilar constructivo de la Tecnología Modular. De esta manera, la definición de un trazado libre de incoherencias, que respete las normativas circulatorias y las restricciones impuestas por el motor gráfico y el módulo de conducción es el primer problema a resolver. Para ello esta Tesis ha desarrollado el módulo de Ajuste Geométrico.

El módulo de Ajuste Geométrico es una herramienta imprescindible en la generación de entornos virtuales destinados a simulaciones de conducción terrestre, ya que es la interpolación de los datos (el modelo a emplear), no solo los datos en sí, lo que determina el éxito de la descripción geométrica base.

Para solventar todas las limitaciones vistas que presentan las aplicaciones SIG comerciales de cara a la generación automática de un simulador de conducción terrestre, esta Tesis ha desarrollado un conjunto de herramientas que han permitido alcanzar la automatización necesaria para este tipo de entornos, respetando siempre las necesidades de los diversos grupos de trabajo involucrados en la simulación.

Se han desarrollado así tres tipos de algoritmos, que se presentan a continuación y que resuelven toda la problemática descrita:

- **Algoritmos de consolidación**: resolverán discontinuidades y discrepancias respecto del trazado topográfico o en su defecto, respecto de puntos de obligado paso.
- **Algoritmos de suavizado**: se encargarán de dotar al trazado de la continuidad necesaria.

- **Algoritmos de gestión de cortes**: como su nombre indica se encargarán de resolver posibles cortes entre trazados.

El desarrollo de estos algoritmos altamente flexibles ha permitido resolver la elevada casuística a tratar, respetando el conjunto de restricciones a las que se ven sometidos estos trazados:

- **Restricciones de la Normativa**: hacen referencia a los radios mínimos, acuerdos y continuidad exigidos a estos trazados para garantizar una conducción cómoda y segura [264] [265] [266].
- **Restricciones del Módulo de conducción:** hacen referencia a las tolerancias de deformación longitudinal necesarias para garantizar un correcto movimiento del vehículo. También pueden hacer referencia al uso de determinados formatos en la información del trazado que se les suministra, con el fin de agilizar sus cálculos. En la presente Tesis se trata de satisfacer las necesidades del módulo de conducción del CITEF, tanto para tráfico ferroviario como urbano [267].
- **Restricciones del Proyecto**: puede requerir el mantener determinadas relaciones geométricas entre elementos, presencia de zonas con geometría predefinida extraída de planos diferentes del topográfico, geometría específica en las intersecciones, etc. En la presente Tesis se han llevado a cabo proyectos de diversa índole como el Ave, Metro de Madrid y Metro de Santiago de Chile. En cada uno de ellos las restricciones fueron distintas. En el último capítulo se presentarán ejemplos de aplicación.
- **Restricciones del motor gráfico**: hacen referencia a la carga poligonal del escenario así como a su estructuración en el grafo de la escena (*scene graph)*. Esto exige una definición geométrica base flexible, que permita una cómoda e intuitiva simplificación y estructuración.

4.4.1 ALGORITMOS DE CONSOLIDACIÓN.

El objetivo de la consolidación es la resolución de las incoherencias existentes entre las diversas fuentes de información suministradas para la representación de un determinado trazado. En la mayoría de los casos, los planos de señalización (Figura 4.23) contienen tanto los datos geométricos del trazado (radios y pendientes) como la información del posicionamiento de elementos decisivos para poder llevar a cabo la simulación (señales, estaciones, conexiones,...). Sin embargo, dicha información geométrica suele presentar discrepancias respecto del trazado topográfico. La coherencia entre ambas fuentes viene garantizada por el empleo de una medida de referencia única: el punto kilométrico (PK). Hay que tener en cuenta que una información topográfica completa del trazado no se tiene y por tanto no sirve como fuente inicial precisa.

[264] Norma UNE-EN 13803-1:2011. Aplicaciones ferroviarias. Vía. Parámetros de proyecto del trazado de la vía. Anchos de vía de 1 435 mm y mayores. Parte 1: Plena vía. Comité AEN/CTN 25 - APLICACIONES FERROVIARIAS

[265] NORMA N.R.V. 0200: Parámetros geométricos. Dirección Técnica de RENFE (Jefatura de Vía); Dirección de Inversiones en Obras e Instalaciones.

[266] http://www.carreteros.org/normativa/trazado/3_1ic/indice.htm. [Última consulta: 11 Enero 2013]

[267] Anexo 2.

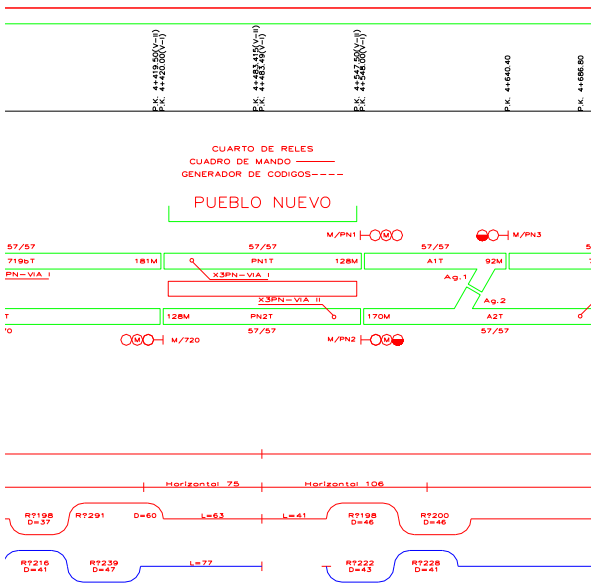


Figura 4.23. Plano esquemático con la información geométrica y de señalización suministrada para la representación de la línea 7 de Metro de Madrid.

La necesidad de mantener una correspondencia biunívoca pk-longitud, obliga al desarrollo de una herramienta que permita generar automáticamente un trazado que represente el plano topográfico lo más fielmente posible, garantizando el exacto posicionamiento de todo elemento del entorno y unas mínimas deformaciones longitudinales respecto de la información suministrada por los diagramas de señalización (Figura 4.24). Con este fin se ha desarrollado la Consolidación.

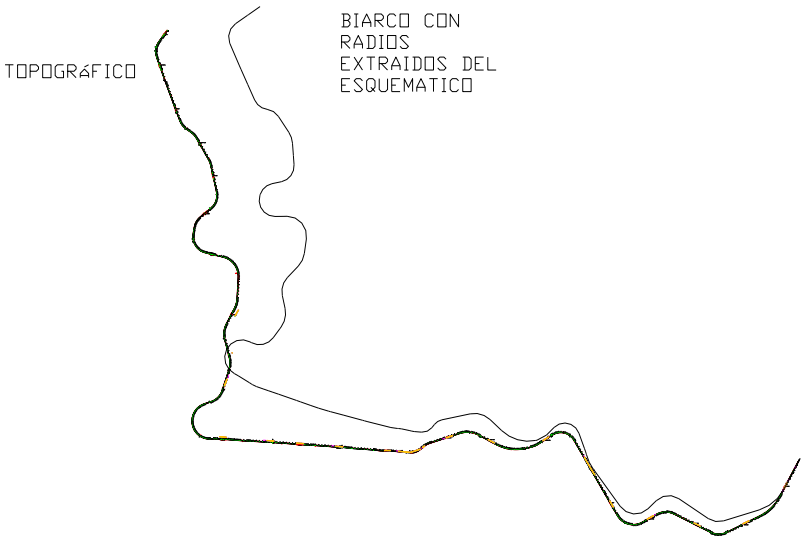


Figura 4.24. Comparación del trazado topográfico y del trazado obtenido a partir de la representación de los datos geométricos suministrados para la línea 7 de Metro de Madrid.

Se define la Consolidación como el proceso que permite aproximar una definición geométrica de entrada (nube de puntos, NURBS o biarco) a una nube de puntos de salida. Debido a las deformaciones que este proceso implica y como consecuencia de la pérdida de identidad en este proceso de los biarcos y NURBS, es necesario transformar estos formatos a uno más

genérico: la nube de puntos. Los datos de entrada así, de un algoritmo de consolidación, son una nube de puntos y un punto final real, extraído por lo general de planos topográficos. Se trata de modificar la nube de puntos de entrada de manera que su punto final coincida con el punto final real, garantizando unas mínimas deformaciones longitudinales y de forma.

Al abordar este problema, esta Tesis planteó diversas opciones. La primera opción consistió en un desplazamiento lineal de todos los puntos de la nube, proporcional a la longitud recorrida. El principal inconveniente de este método era la gran discontinuidad tangencial producida en los extremos del intervalo. Si bien un posterior suavizado podía resolver esta discontinuidad, esto era a costa de una mayor deformación de la nube.

La segunda opción se basó en un desplazamiento cúbico de los puntos proporcional a la longitud recorrida. El principal inconveniente de este método fue la introducción de un mayor número de oscilaciones, produciéndose así una gran distorsión de la forma.

Los siguientes planteamientos fueron los que proporcionaron la solución satisfactoria y consistieron en un escalado y giro de la nube de puntos. Fueron dos las variantes: escalado en XY y escalado en X. Como a continuación se verá, dependiendo del tipo de error de datos que sea necesario solventar será más idónea una u otra.

Dada una nube de puntos de entrada $\Gamma(x,y)$, sean $P_0\ (x_0, y_0)$ y $P_f\ (x_f, y_f)$ las coordenadas cartesianas de su punto de fin y del punto de fin real extraído del topográfico respectivamente, referidas al pivote de la consolidación (el punto de inicio de la nube), P_v.

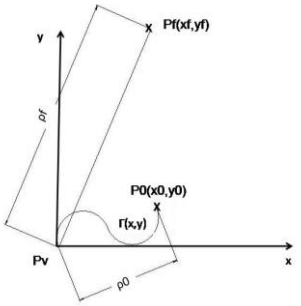


Figura 4.25. Ejemplo de consolidación de un biarco.

Los pasos a seguir por el algoritmo de Consolidación son los siguientes. Se gira $\Gamma(x,y)$ hasta hacerla coincidir con el eje X, obteniendo así la curva $\Gamma_1(x,y)$:

$$\Gamma_1(x,y)= \Gamma(x,y)\begin{pmatrix} \frac{x_0}{\rho_0} & \frac{-y_0}{\rho_0} \\ \frac{y_0}{\rho_0} & \frac{x_0}{\rho_0} \end{pmatrix} = \Gamma(x,y)\, M_0^{-1}$$

A continuación se escala Γ_1 hasta que la distancia entre el punto origen y el punto fin de la línea (inicialmente ρ_0) pase a ser ρ_f (Figura 4.26) , distancia entre el punto de inicio y de fin en la trayectoria real (extraída del plano topográfico). En el caso de escalado en X:

$$\Gamma_2(x,y)= \Gamma_1(x,y)\begin{pmatrix} \frac{\rho_f}{\rho_0} & 0 \\ 0 & 1 \end{pmatrix} = \Gamma_1(x,y)\, M_{es}$$

En el caso de escalado en XY:

$$\Gamma_2(x,y)= \Gamma_1(x,y)\begin{pmatrix} \frac{\rho_f}{\rho_0} & 0 \\ 0 & \frac{\rho_f}{\rho_0} \end{pmatrix} = \Gamma_1(x,y)\, M_{es}$$

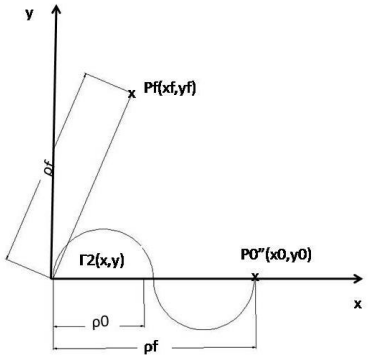


Figura 4.26. Ejemplo de consolidación de un biarco: giro y escalado XY.

Finalmente se gira la trayectoria de modo que el punto final coincida con el de la trayectoria real, obteniendo la curva consolidada Γ_3 :

$$\Gamma_3(x,y)= \Gamma_2(x,y)\begin{pmatrix} \frac{x_f}{\rho_f} & \frac{y_f}{\rho_f} \\ \frac{-y_f}{\rho_f} & \frac{x_f}{\rho_f} \end{pmatrix} = \Gamma(x,y)\, M_0^{-1}\ M_{es}\ M_f$$

La Figura 4.27 muestra los resultados ofrecidos por los diferentes algoritmos de consolidación aplicados al biarco de partida representado en negro, para un punto final real P_F.

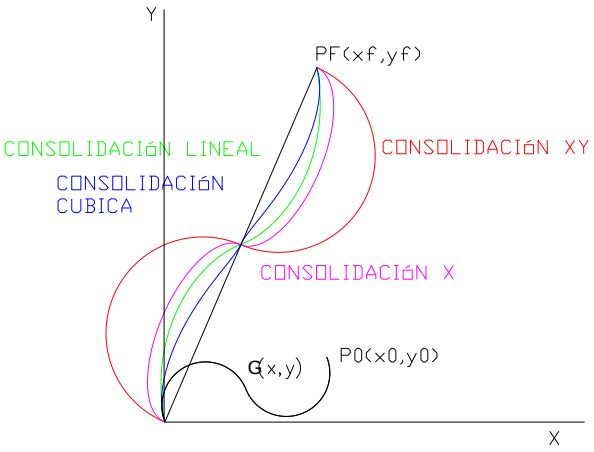
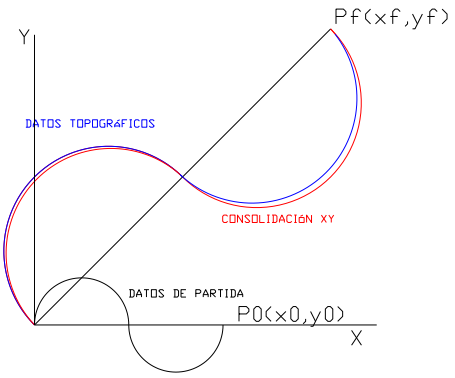
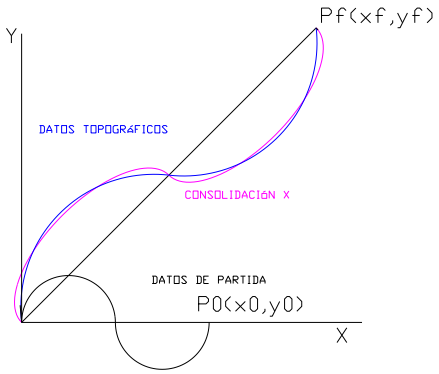


Figura 4.27. Ejemplos de diversos tipos de consolidaciones de un biarco.

Como se puede observar el algoritmo de consolidación lineal presenta una gran distorsión en la tangente de salida y la consolidación cúbica una excesiva oscilación, lo que lleva a desestimar ambas soluciones. La elección entre la solución ofrecida por el escalado en X o el escalado XY dependerá de la deformación longitudinal sufrida por los datos de entrada y el grado de aproximación a los datos topográficos disponibles. La Figura 4.28 muestra dos ejemplos en los que es necesaria una consolidación de los datos iniciales: dada una definición geométrica de partida (biarco negro), unos datos topográficos (nube de puntos azul) y un punto final real Pf, las soluciones más idóneas serán las ofrecidas por la Consolidación XY en el caso (a) y la Consolidación X en el caso (b).



(a)



(b)

Figura 4.28. Consolidación idónea: (a) Consolidación XY.(b) Consolidación X.

La evaluación de la deformación longitudinal y la conservación de la forma se realizarán tras el proceso de suavizado. En ese momento se optará por la consolidación X o XY.

4.4.2 ALGORITMOS DE SUAVIZADO.

El objetivo de este conjunto de algoritmos es aproximar una nube de puntos mediante una curva capaz de cumplir las especificaciones que pueden ser exigidas en simulaciones de conducción terrestre:

- Radios y pendientes dentro de los intervalos establecidos.
- Continuidad tangencial y de curvatura.
- Deformación longitudinal dentro de las tolerancias admisibles por el proyecto.
- Posibilidad de reconocimiento del trazado por parte del conductor, en el caso de representación de entornos reales.
- Empleo de un formato compatible con :
    - el módulo de conducción: ha de favorecer la agilización de sus cálculos.
    - el motor gráfico: es aconsejable un formato que permita una cómoda e intuitiva simplificación poligonal.

Con el fin de satisfacer todos estos requisitos se han seleccionado dos posibles formatos: el biarco y la NURBS.

BIARCOS:

Se define una curva biarco (*arc spline*) como una curva formada por la concatenación de biarcos simples garantizando en su unión una continuidad de la derivada o continuidad C^1.

Dados dos arcos circulares (que pueden degenerar en un único arco o recta) A_0 y A_1, se dice que constituyen un biarco interpolador simple de los datos orientados C^1, representados por los puntos extremos P_0, P_1 y los vectores tangentes unitarios U_0, U_1, si y solo si los dos arcos comparten un punto extremo J, llamado punto de unión y satisfacen las siguientes propiedades:

- El arco A_0 tiene como extremos los puntos P_0 y J y U_0 es tangente a A_0 con la orientación correspondiente a la parametrización de A_0 desde P_0 a J.
- El arco A_1 tiene como extremos los puntos J y P_1 y U_1 es tangente a A_1 con la orientación correspondiente a la parametrización de A_1desde J a P_1.
- Los dos arcos comparten un vector unitario tangente en su punto de unión J, con la orientación correspondiente a una parametrización de A_0 desde P_0 a J y de A_1, desde J a P_1.

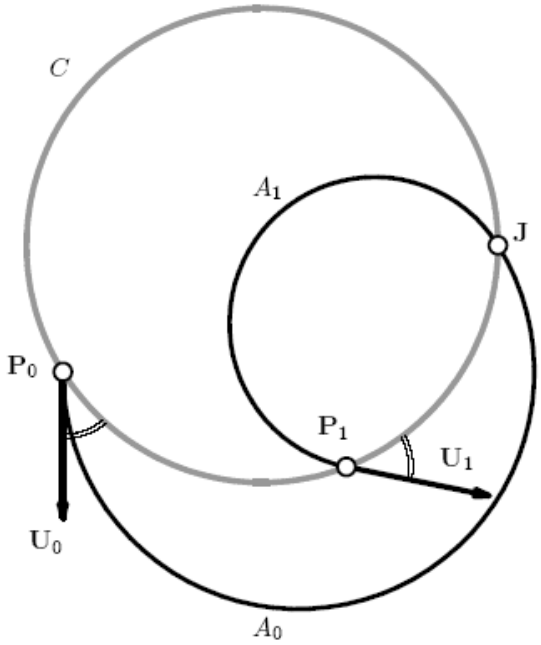


Figura 4.29. Biarco simple interpolador de los datos: P_0, P_1, U_0, U_1.

Las curvas biarco presentan la ventaja de ofrecer una cómoda manipulación, permitiendo conocer y controlar la curvatura en cualquier punto de la misma mediante un número finito de parámetros y establecer fácilmente relaciones de paralelismo, ya que la paralela a una curva biarco es otra curva biarco, a diferencia de las B-splines[268]. Las curva biarco presentan múltiples aplicaciones: máquinas de CN [269] [270], prototipado rápido [271], determinación de trayectorias de robots [272] [273] o el trazado de obras lineales [274] [275] [276], caso en el que se centra este apartado.

A pesar de su condición C^1, el biarco no supone limitación alguna a la hora de representar trazados ferroviarios o urbanos, ya que el salto de curvatura puede hacerse tan pequeño como se desee, basta con tomar el número suficiente de arcos [277].

NURBS:

Dentro de las Splines, la presente Tesis ha seleccionado la NURBS por ser la que presenta la máxima generalización y por tanto la que permite la mayor flexibilidad, destacando por encima de todo su capacidad de representar curvas racionales. A lo largo de las dos últimas décadas las NURBS (Non Uniform Rational B-Splines), han adquirido gran relevancia en el diseño geométrico y han sido incorporadas en diversas aplicaciones comerciales dedicadas al modelado. Pero si bien, la versatilidad que se consigue mediante la introducción de modificaciones en el polígono de control, en los pesos y en el vector de nodos es muy elevada, el gran hándicap lo constituye lo poco intuitivo de su manejo así como la ambigüedad que ofrecen, ya que una misma forma geométrica puede describirse mediante diferentes combinaciones del polígono de control, vector de nodos y pesos. Esta redundancia geométrica convierte la simplificación, algo muy necesario en toda representación virtual, en un proceso ad hoc y confuso. A continuación se muestra una comparativa sobre las ventajas e inconvenientes ofrecidos por cada una de las definiciones geométricas base comentadas:

[268] Elber, G., Lee, I., Kim, M. Comparing Offset Curve Approximation Methods. IEEE Computer Graphics and Applications 1997. Vol.17, n. 3, pp 62-71.

[269] Yeung, M.K., Walton, D.J. Curve fitting with arc splines for NC toolpath generation. Computer-Aided Design 1994. pp 845-849.

[270] Qiu, H., Cheng, K. and Li, Y. Optimal circular arc interpolation for NC tool path generation in curve contour manufacturing. Computer-Aided Design 1997. Vol. 29, n. 11, pp. 751–760.

[271] Koc, B., Ma, Y., Lee, Y. Smoothing STL Files by Max-t Biarc Curves for Rapid Prototyping. Rapid Prototyping Journal 2000. Vol. 6, n. 3, pp 186-203.

[272] Laumond, J.P, Jacobs, P., Taix M, Murray R. A motion planner for non-holonomic mobile robots. IEEE Transactions on Robotics and Automation 1994. Vol.10, n. 5, pp. 577–593.

[273] Yang, X. Efficient circular arc interpolation based on active tolerance control. Computer Aided Design 2002. Vol. 34, n. 13, pp. 1037–1046.

[274] Baass, KG. The use of clothoid templates in highway design. Transportation Forum 1984. Vol.1, pp 47–52.

[275] Walton, D.J, Meek, D.S. Computer-aided design for horizontal alignment. ASCE Journal of Transportation Engineering 1989. Vol. 115, n. 4, pp. 411–24.

[276] Walton, D.J, Meek, D.S. Clothoidal splines. Computers & Graphics 1990. Vol. 14, n. 1, pp. 95–100.

[277] Walton, D.J, Meek, D.S. An arc spline approximation to a clothoid. Journal of Computational and Applied Mathematics 2004. Vol. 170, n. 1, pp. 59-77.

NURBS	**Biarcos**
En el caso de nubes con gran cantidad de puntos su procesamiento implica un alto coste computacional.	Buen comportamiento en entornos de gran tamaño.
El cálculo de puntos, radios de curvatura, paralelas..etc se complica, siendo frecuente la resolución de ecuaciones no lineales.	Simplicidad de la curva solución: Cálculo inmediato de tangentes, radios, centros de curvatura..(útil para determinar gran cantidad de magnitudes:velocidades, distancias,...)
A pesar de permitir una gran flexibilidad (se permite un control puntual de la curvatura), la dificultad para evaluar su valor exacto en cada punto hace muy costoso en estos proyectos respetar los límites impuestos por la normativa del trazado.	A pesar de permitir una menor flexibilidad que la NURBS (no permiten el control puntual de la curvatura), los errores de curvatura que introducen son generalmente reducibles al intervalo de tolerancias admisibles por este tipo de proyectos.
Estable frente a modificaciones en la nube de partida.	Muy sensible a pequeños cambios en la nube de partida (que en esta Tesis se han minimizado)
Alto grado de aproximación incluso en zonas pequeñas.	Peor aproximación.
Permite funciones continuas hasta el orden n, siendo n el grado de la NURBS.	Devuelve una curva continua de primer orden, sin embargo, mediante la introducción de aproximaciones de la clotoide se puede simular la continuidad hasta el segundo orden.
Buena aproximación de las curvas de transición.	Cálculo directo de curvas clotoide [278] [279]

Analizando los posibles requisitos exigibles a estos trazados y las aportaciones que ambos tipos de definiciones geométricas pueden ofrecer, se establece como conclusión que el biarco se adapta mejor a las necesidades de un sistema de simulación de conducción terrestre, por lo que será la definición geométrica base que se utilice en esta Tesis. Sin embargo, ante la posibilidad de que existan zonas conflictivas que hayan de reproducir con una tolerancia muy restrictiva un determinado tramo del trazado, se permitirá la combinación del biarco con intervalos de NURBS.

En la presente Tesis se abordará el problema del suavizado mediante la distinción entre dos clases de algoritmos:

- **Algoritmos de pequeña escala**: plantean la resolución del problema de generación de un biarco simple cumpliendo las llamadas condiciones de Hermite (pasar por dos puntos

[278] Walton, D.J, Meek, D.S. An arc spline approximation to a clothoid. Journal of Computational and Applied Mathematics 2004. Vol. 170, n. 1, pp. 59-77.

[279] Wang, L.Z., Miura, K.T., Nakamae, E., Yamamoto T., Wang, T.J. An approximation approach of the clothoid curve defined in the interval [0; π/2] and its offset by free-form curves. Computer Aided Design 2001. Vol. 33, pp. 1049–1058.

extremos y ser tangente en estos puntos).

- **Algoritmos de gran escala:** plantean el problema del suavizado de un determinado trazado mediante una curva biarco (*arc spline*) a través de sucesivas llamadas a los algoritmos de pequeña escala.

4.4.2.1 ALGORITMOS DE PEQUEÑA ESCALA.

El problema planteado anteriormente para la determinación del biarco simple, cumpliendo las llamadas condiciones de Hermite presenta infinitas soluciones. El lugar geométrico del punto de unión de ambos arcos constituyentes del biarco simple es un círculo que pasa por ambos puntos extremos.

Con el fin de obtener, a partir de dicho lugar geométrico, el punto de cambio de curvatura del biarco que lo haga único, en la literatura se han planteado varias condiciones adicionales: minimización de la diferencia de ambos radios [280], cociente de radios próximo a uno[281], minimización de la diferencia de curvaturas [282] y paralelismo de la tangente intermedia a una determinada dirección [283] son las más empleadas.
En la presente Tesis se pretende generar tanto trayectorias con sentido ferroviario, como trayectorias con sentido urbano e interurbano y las restricciones a las que se somete al biarco simple son las siguientes:

- Los arcos tendrán que superar un radio mínimo, que vendrá impuesto por la normativa de trazado correspondiente o por el propio proyecto.

- La longitud del biarco deberá encontrarse dentro de las tolerancias de deformación longitudinal impuestas por el módulo de conducción.

Estas restricciones han motivado el desarrollo de una nueva metodología de cálculo que seleccione de entre todos los biarcos posibles, aquél que se ajuste mejor a las necesidades existentes. Esta metodología se basa en la parametrización de uno de los radios del biarco en función del otro. En primer lugar se determinará el intervalo de radios posibles para la solución y posteriormente se seleccionarán los que mejor aproximen la longitud deseada.

CALCULO DEL BIARCO SIMPLE:

El método desarrollado en esta Tesis resuelve la interpolación de Hermite de un biarco simple (Figura 4.30), garantizando que su longitud se encuentre dentro de un determinado intervalo de tolerancias y que el radio de ambos arcos sea superior a un radio mínimo dado. Cuando estás condiciones no se pueden cumplir se genera una solución tri-arco o bien se divide en dos biarcos.

[280] Bolton, K M. Biarc curves. Computer Aided Design 1975. Vol. 7, pp. 89-92.

[281] Sabin, M.A. The use of piecewise forms for the numerical representation of the shape. Report n. 60. Computer and Automation Institute, Hungarian Academy of Sciences 1977.

[282] Nutbourne, A. W. and Martin R. R. Differential Geometry Applied to Curve and Surface Design. Vol. 1: Foundations Ellis Horwood, Uk. 1988. ISBN: 978-0470210369.

[283] Sharrock, T. J. Biarcs in three dimensions. In Proceedings on Mathematics of Surfaces II . Ed. R. R. Martin, Oxford Univ. Press, New York, 1988. pp. 395-411.

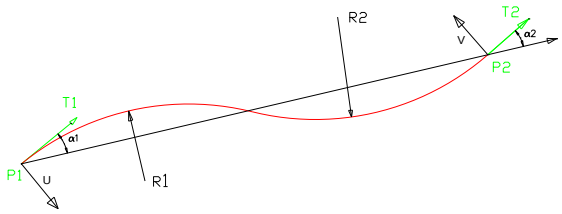


Figura 4.30. Biarco simple.

El diagrama de flujo del algoritmo es el siguiente:

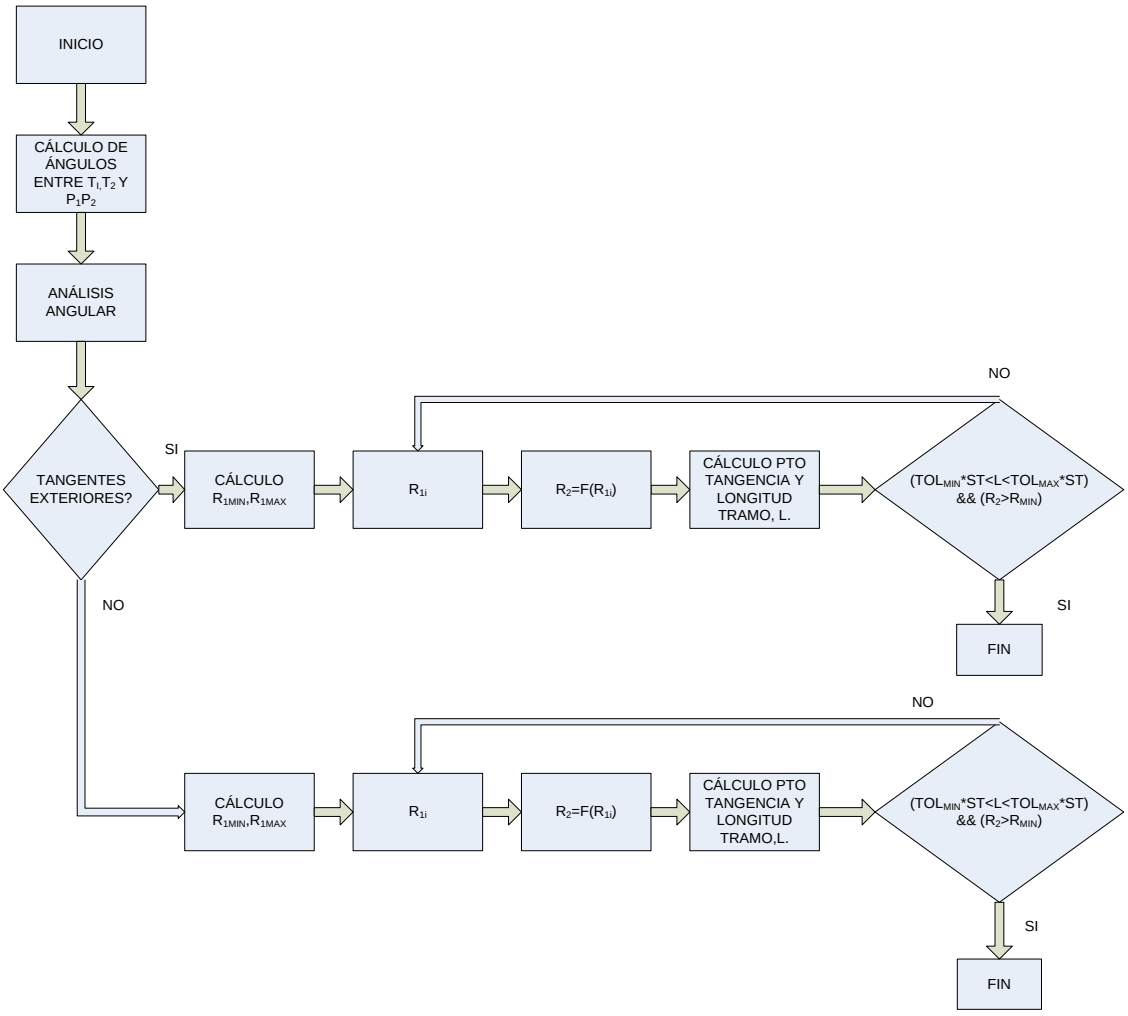


Figura 4.31. Diagrama de flujo para el cálculo de un biarco simple.

Sean:

- P_1 y P_2, los puntos inicial y final respectivamente,
- T_1 y T_2 las tangentes inicial y final respectivamente.

- R_1 y R_2 los radios de los dos arcos que forman el biarco.
- Tol_min: la tolerancia de acortamiento por unidad de longitud.
- Tol_max: la tolerancia de alargamiento por unidad de longitud.
- R_{min}, el radio mínimo exigido.
- ST: longitud de biarco deseada (la extraída de los planos de señalización).

En primer lugar se hallan los ángulos relativos de las tangentes T_1 y T_2 respecto de la recta P_1P_2. Con el valor de estos ángulos relativos se realiza un análisis angular. De este análisis se concluye si las tangentes son exteriores o interiores. Si T_1 y T_2 se encuentran en el mismo semiplano con respecto a la recta P_1P_2 las circunferencias resultantes serán tangentes exteriores. Si por el contrario T_1 y T_2 se encuentran en semiplanos distintos serán tangentes interiores.

Una vez determinado el tipo de tangencia se parametriza R_2 en función de R_1, obteniendo las siguientes expresiones:

- si se trata de circunferencias tangentes exteriores:

$$R_2 = \frac{-\left(P_{2x} - P_{1x} - R_1 * U_x\right)^2 - \left(P_{2y} - P_{1y} - R_1 * U_y\right)^2 + R_1^{\,2}}{2 * Vx\left(P_{2x} - P_{1x} - R_1 * U_x\right) + 2 * Vy\left(P_{2y} - P_{1y} - R_1 * U_y\right) - 2 * R_1}$$

- si se trata de circunferencias tangentes interiores:

$$R_2 = \frac{-\left(P_{2x} - P_{1x} - R_1 * U_x\right)^2 - \left(P_{2y} - P_{1y} - R_1 * U_y\right)^2 + R_1^{\,2}}{2 * Vx\left(P_{2x} - P_{1x} - R_1 * U_x\right) + 2 * Vy\left(P_{2y} - P_{1y} - R_1 * U_y\right) + 2 * R_1}$$

Donde:

- $P_1(P_{1x}, P_{1y})$ son las coordenadas cartesianas del punto inicial.
- $P_2(P_{2x}, P_{2y})$ son las coordenadas cartesianas del punto final.
- $U(U_x, U_y)$ es el vector director unitario de la recta perpendicular a T_1 en dirección al centro de la circunferencia C_1 .
- $V(V_x\ V_y)$ es el vector director unitario de la recta perpendicular a T_2 en dirección al centro de la circunferencia C_2.
- R_1 radio de la circunferencia tangente a T_1.
- R_2 radio de la circunferencia tangente a T_2,

siendo el criterio de determinación del sentido de los vectores el indicado en la Figura 4.30.

Estudiando el comportamiento de estas funciones se llega a las siguientes conclusiones:

- Presentan una asíntota vertical para:
  - $R_1 = \frac{V * d}{U * V + 1}$ si son tangentes exteriores;
  - $R_1 = \frac{V * d}{U * V - 1}$ si son tangentes interiores,

donde d es el vector director de la recta P_1P_2.

En el caso de tangentes exteriores el denominador siempre va a ser positivo independientemente de la orientación de los ángulos (U*V $\in$ (-1,1)) y dado el convenio de sentidos que se ha tomado (R1 lleva incorporado el signo en la ecuación por tanto R1>0), los valores para los cuales V*d < 0 están fuera del campo de validez de la solución, es decir, V y d forman un ángulo entre -90 y 90 grados. O lo que es lo mismo, si d y T_1 forman entre 90 y 270 grados.
Para el caso de circunferencias tangentes interiores, la solución será valida (R_1>0) si V y d forman un angulo entre 90 y 270 grados. O lo que es lo mismo, si d y T_1 forman entre -90 y 90 grados.

- Presentan una asíntota horizontal para $R_1 \longrightarrow \pm\infty$:
  - $R_2 = \dfrac{U * d}{-U * V - 1}$ si son tangentes exteriores ;
  - $R_2 = \dfrac{U * d}{-U * V + 1}$ si son tangentes interiores.

- Son funciones monótonas, decreciente en el caso de tangentes exteriores y creciente en el caso de las interiores.

La Figura 4.32 muestra un gráfico que describe el comportamiento de R_2 en función de R1 para el caso de unión mediante arcos tangentes exteriores para los valores:

•P_1 (50,0).
•P_2 (500,300).
•T_1 forma 1.5 radianes con el eje x.
•T_2 forma 0.9 radianes con el eje x.

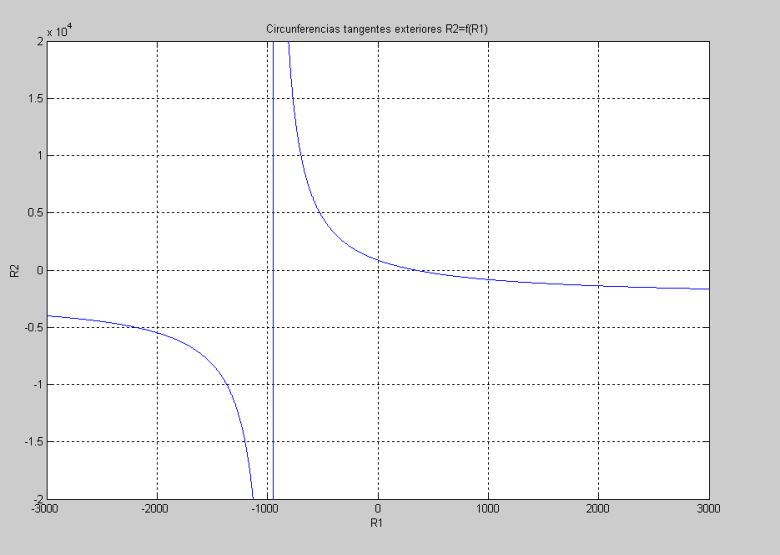


Figura 4.32. Lugar geométrico de los radios para circunferencias tangentes exteriores.

La Figura 4.33 describe el comportamiento de R_2 en función de R_1 para el caso de unión mediante arcos tangentes interiores para los valores:

•P_1 (50,300).
•P_2 (500,300).
•T_1 forma 1.5 radianes con el eje x.
•T_2 forma 5.5 radianes con el eje x.

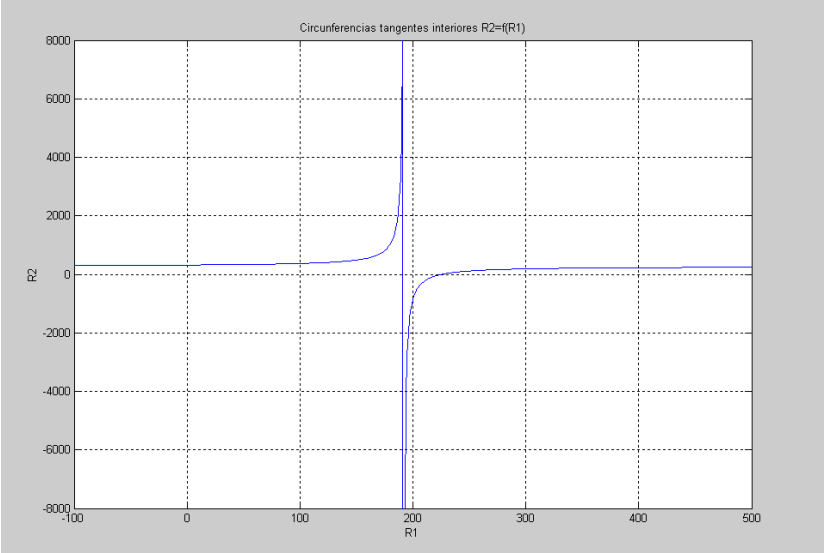


Figura 4.33. Lugar geométrico de los radios para circunferencias tangentes interiores.

A continuación se calcula el valor máximo de R_1 en función del tipo de tangencia:

- Tangentes exteriores: El valor máximo es el asociado al R_2 mínimo (monótona decreciente).

- Tangentes interiores: En este caso por un lado se tendrá que tener en cuenta si se encuentra antes o después de la asíntota vertical. En el primer caso el R_1 máximo será el asociado a la asíntota y en el segundo, con el fin de no alargar demasiado el proceso de cálculo se calculará el R_1 asociado a un R_2 ligeramente inferior al valor de la asíntota horizontal.

Por último se hace un barrido de radios con $R_1 \in$[R_1 mínimo, R_1 máximo]. En el caso de que exista un radio R_1 deseado se partirá de éste (caso que ocurrirá tras la consolidación topográfica de la información geométrica procedente de un plano de señalización, siendo el valor del radio R_1 deseado el extraído de dicho plano), en caso contrario el barrido comenzará tomando R_1 igual al valor medio del intervalo [R_1mínimo, R_1 máximo]. A continuación se halla el punto de tangencia de los arcos y la longitud del tramo generado para cada par (R_1, R_2). En el caso de que exista solución el algoritmo la devuelve. En caso de no existir solución el algoritmo solicita la modificación de alguno de los parámetros de entrada y se repite el mismo proceso de cálculo hasta encontrar una solución. Este proceso puede exigir la división del tramo P_1P_2 en nuevos subintervalos, para cada uno de los cuáles se repetirá el proceso de búsqueda de solución.

4.4.2.2 ALGORITMOS DE GRAN ESCALA.

Se trata de aproximar una nube de puntos (que surge de un proceso de consolidación o de la lectura de un plano topográfico) mediante una curva biarco (*arc spline*), de tal modo que el resultado sea una trayectoria correspondiente a una línea ferroviaria, urbana o interurbana. La

trayectoria resultante debe mantener la forma de la nube de puntos en la medida de lo posible y respetar a su vez unas restricciones de longitud y de radio mínimo.

Como datos de partida tenemos pues una nube de puntos, la longitud de trayectoria deseada, ST, una tolerancia de deformación longitudinal mínima, tol_min, una tolerancia de deformación longitudinal máxima, tol_max y un radio mínimo, r_min.

El primer paso a seguir en un algoritmo de suavizado es la determinación de los puntos de interpolación o vector de nodos y la aproximación de la tangente en estos puntos. Para la selección de dichos nodos se han formulado varios algoritmos entre los que destacan los de Meek y Walton [284]:

- **Método del Mayor Arco:** El primer punto de la nube será el primer nodo. A continuación se busca el biarco que aproxime el mayor número de puntos posible respetando las tolerancias dadas. El siguiente nodo será el punto final de este biarco y así sucesivamente.
- **Método de la Bisección:** En primer lugar se construirá el biarco entre el primer y el último punto de la nube dada y se irá fraccionando esta selección hasta que se logre el cumplimiento de las tolerancias deseadas.

Estudios posteriores presentan optimizaciones de estos métodos minimizando el número de arcos y el tiempo de cálculo [285] [286].

En la presente Tesis, con el fin de respetar al máximo la forma de la nube y las tolerancias exigidas, se ha optado por emplear como vector de nodos, el de puntos aproximados de cambio de curvatura. Su cálculo variará en función de los datos de partida de que se dispongan:

- **Suavizado de una nube de puntos obtenida tras un proceso de consolidación**, caso que generalmente se presenta en la representación de líneas ferroviarias, donde se dispone de información topográfica y planos esquemáticos con información geométrica y de señalización. En este caso cada punto llevará asociado un radio de partida, extraído del plano de señalización.
- **Suavizado de una nube de puntos extraída directamente del topográfico**, caso que generalmente se presenta en la generación de ciudades, donde la única información disponible son los planos topográficos. En este caso para determinar los puntos de cambio de curvatura será necesario calcular una aproximación del radio en cada punto de la nube. Para ello se detectarán en primer lugar las rectas existentes en la nube de puntos. Si el coeficiente de correlación de la recta de mínimos cuadrados supera el test con una precisión de una parte por mil, se considera una recta. A continuación, para todos los puntos pertenecientes a los intervalos entre rectas, se evaluará cada tres puntos el arco que pasa por ellos y se adjudicará dicho radio al punto central. Los puntos de cambio de curvatura serán aquellos puntos cuyo radio no exceda respecto de su radio anterior una determinada tolerancia. Esta tolerancia se ha determinado heurísticamente, empleando para ello criterios perceptivos. En entornos ferroviarios esta tolerancia se ha establecido en un 20% y en entornos urbanos en un 10%.

Para el problema de aproximación de la tangente en los puntos pertenecientes al vector de nodos existen gran cantidad de soluciones [287]. Walton y Xu [288] la aproximan como el ángulo

[284] Meek, D.S and Walton, D.J. Approximation of discrete data by G1 arc splines. Computer-Aided Design, 1992. Vol. 24, n. 6. pp. 301-306.

[285] Held, M., Eibl, J. Biarc Approximation of Polygons Within Asymetric Tolerance Bands. Computer-Aided Design, 2005. Vol. 37, n. 4., pp. 357-371.

[286] Yang, X. Approximating NURBS Curves by Arc Splines. In Proceedings of the Geometric Modeling and Processing 2000. IEEE Computer Society Press. pp. 175-183.

formado por el vector que une el punto inmediatamente anterior y el posterior. Shippey [289] la define como la tangente en el punto del círculo que pasa por este y los dos más próximos a él. Renner [290] da dos aproximaciones en el punto medio de cinco puntos. En esta Tesis se ha optado por utilizar la solución elegida por Walton y Xu empleando como puntos de interpolación la nube de puntos a suavizar. Se ha optado por esta solución, porque la nube de puntos inicial con la que se trabaja es siempre lo suficientemente densa, como para ofrecer en todos los casos resultados satisfactorios.

El algoritmo de suavizado es un algoritmo recursivo que recorre el vector de nodos calculando un biarco simple por cada par de ellos (N_i, N_{i+1}). En cada paso iterativo se calcula la variación de longitud sufrida por la curva biarco, L, y en caso de no verificar las tolerancias de deformación longitudinal del proyecto (TOL_{max}, TOL_{min}), la nube de puntos se desplaza proporcionalmente a esta variación de longitud sufrida. Se repite este proceso hasta encontrar la longitud que verifica las tolerancias de deformación longitudinal del proyecto (Figura 4.34).

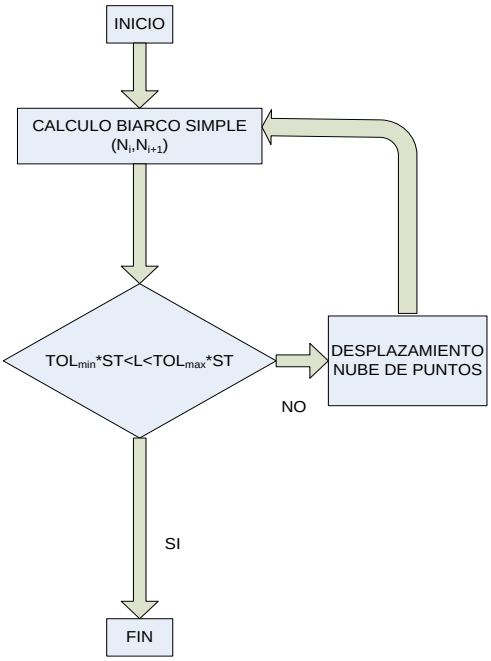


Figura 4.34. Algoritmo de suavizado para los nodos (Ni, Ni+1).

Los puntos de la nube se desplazan de modo tal que se tienda a aumentar su radio de curvatura en el caso de que tengamos que alargar la línea o disminuirlo, si tenemos que encogerla. Para ello tomamos como punto de partida los arcos del paso anterior de iteración y movemos los puntos a partir de los mismos. En la siguiente figura se observa el desplazamiento de uno de los

[287] Meek, D.S, Walton D.J. Several methods for representing discrete data by line segments. Cartographica 1991.Vol. 28, pp 13-20.

[288] Walton, D.J and Xu, R. Turning point preserving planar interpolation. ACM Transactions Graphics 1991. Vol. 10, n. 3, pp. 301-302.

[289] Shippey, G A. Interpolation through a set of data points using circular arc segments. Internal Note Tech/IEG/67/15 Ferranti, Edinburgh,UK. 1967.

[290] Renner, G A. A method of shape description for mechanical engineering practice. Computers in Industry, 1982. Vol 3, pp. 137-142.

puntos de la nube, P_old , con el fin de alargar la longitud del biarco. El nuevo punto desplazado, P_new, se obtiene a partir del punto inicial, P_old , llevando una distancia ,*despl*, sobre el radio del arco anterior. En la siguiente iteración el arco pasará por P_new.

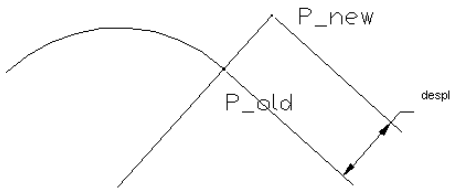


Figura 4.35. Desplazamiento de la nube de puntos.

Para determinar el desplazamiento, se hace la siguiente suposición:

$$error = |ST - L_1| = \sum (R_{2i} * \alpha_{2i} - R_{1i} * \alpha_{1i}) = despl * \sum \alpha_i$$

$R_{2i} \approx R_{1i} + despl$

$\alpha_{2i} \approx \alpha_{1i}$

siendo:

S_T*: longitud de la línea deseada.*

L_1*: la longitud de la línea en el último paso (conocida).*

R_{2i}*:el radio del arco i deseado (no conocido).*

R_{1i}*: el radio del arco i en el último paso (conocido).*

α_{2i} *: el ángulo del arco i deseado (no conocido).*

α_{1i} *: el ángulo del arco i en último paso (conocido).*

De esta manera se tiene que el desplazamiento es igual al error cometido dividido por la suma de los ángulos de los arcos de circunferencia.

$$despl = \frac{error}{\sum \alpha_i}$$

Para proyectos donde la restricción de similitud con la realidad no sea muy severa, y con el fin de hacer el algoritmo más flexible y permitir así encontrar un mayor abanico de soluciones posibles, se ha desarrollado una variante del algoritmo de suavizado que permite la introducción de una tolerancia de deformación angular en la tangente para todos los puntos del vector de nodos excepto en los extremos, donde la necesidad de mantener intactas las zonas limítrofes al intervalo de suavizado es una premisa de todo proyecto.

4.4.3 ALGORITMOS DE GESTIÓN DE CORTES.

El objetivo de esta metodología es corregir automáticamente los cortes indeseados que suelen producirse entre las líneas que configuran un escenario, al emplear como información de partida la definición geométrica y las restricciones topológicas suministradas por los planos de señalización.

El algoritmo de gestión de cortes diseñado por esta Tesis actúa por parejas de puntos de corte (Figura 4.36). La nueva descripción geométrica ofrecida por el algoritmo en cada uno de

estos tramos, (*zonas de corte)* deberá encontrarse dentro del intervalo de tolerancias longitudinales admitido para el proyecto.

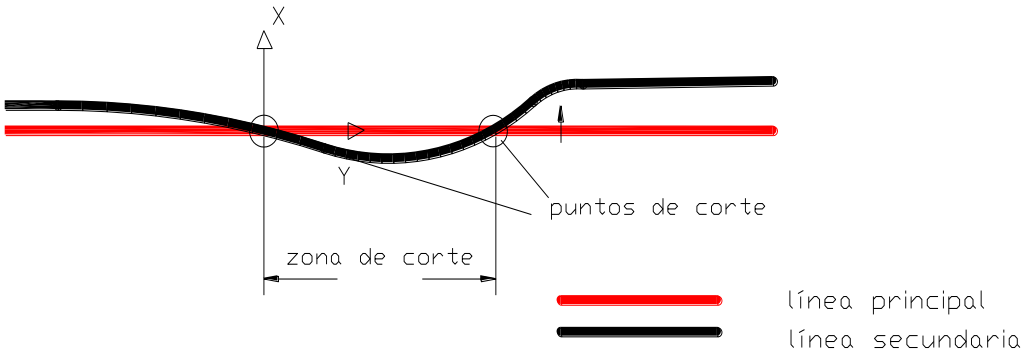


Figura 4.36. Ejemplo de dos líneas que se cortan por error de datos de entrada.

La importancia de la longitud a la hora de definir el trazado ha motivado el desarrollo de dos métodos:

- El primer método, *algoritmo de traslación*, se ha creado con el fin de conservar al máximo la forma de la línea. Consiste en trasladar la zona comprendida entre los puntos de corte de la línea invasora (línea de menor prioridad constructiva [291]), una distancia que asegure que las líneas no vuelvan a intersectarse en ese tramo (*distancia máxima de traslación)* y que se respeten las distancias mínimas de separación entre líneas.

- El segundo método, *algoritmo de aplanamiento*, se utilizará en los casos en los que la longitud de la nueva zona calculada mediante el primer método no esté dentro de la tolerancia permitida. En este caso en la zona de corte no se traslada la geometría, sino que se sitúan los puntos en una recta paralela a la *recta de referencia (*recta que une los puntos de corte*)*, a una distancia igual a la *distancia máxima de traslación*.

El algoritmo de gestión de cortes comienza considerando la jerarquía de prioridades constructivas existente entre líneas (Figura 4.37), ya que de lo contrario es posible que se produzcan nuevos cortes en zonas ya resueltas. De esta manera, al resolver un corte entre dos líneas, siempre se editará la de menor prioridad.

Una vez identificada según este criterio la línea a evaluar, el primer paso a seguir será el cálculo de los puntos de corte. Los datos de partida nos dan información de la traza media de las líneas. Para calcular los cortes, en lugar de utilizar esta información, se van a determinar las paralelas más desfavorables a éstas, teniendo en cuenta para ello la posición relativa de las líneas en estudio y la distancia mínima a la que se pueden encontrar éstas en esta zona (parámetro de diseño).

El algoritmo de gestión de cortes actúa, como ya se ha comentado, por parejas de puntos de corte, generándose las llamadas *zonas de corte*. Para deshacer el corte en cada una de estas zonas, se comienza aplicando el *algoritmo de traslación*.

[291] En el apartado 4.5.2. se explicarán estas prioridades constructivas.

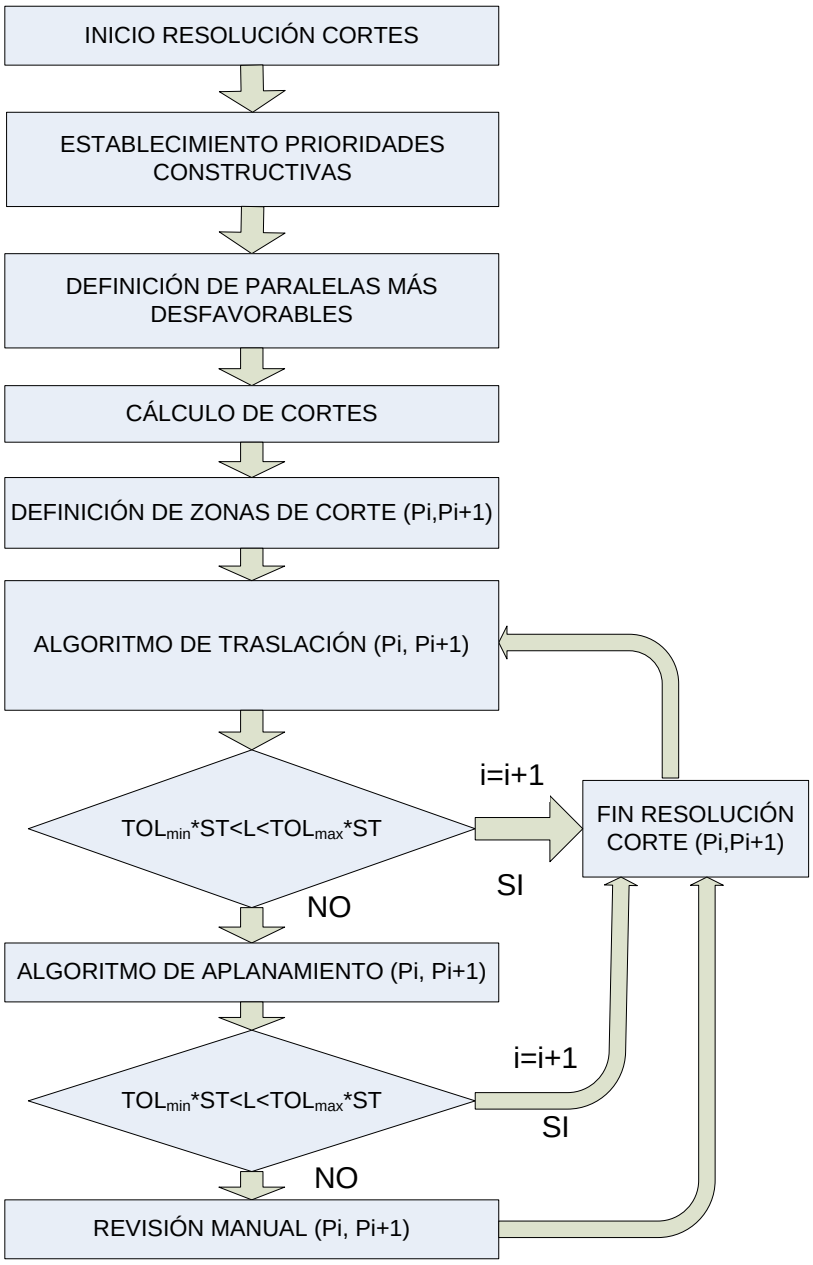


Figura 4.37. Algoritmo de gestión de cortes.

El *algoritmo de traslación* (Figura 4.38) traslada la nube de puntos en la dirección perpendicular a la *recta de referencia* una distancia llamada *distancia máxima de traslación*. Se ha optado por una traslación porque es un método que no deforma la geometría de la zona afectada. El sentido de traslación dependerá de la posición relativa entre las líneas.

La distancia máxima de traslación se calculará de manera que se garantice que no se vuelva a producir un corte en esta zona. Para facilitar y acelerar el proceso de cálculo, se realiza un cambio de coordenadas donde el nuevo origen será el punto de corte inicial y los nuevos ejes estarán definidos por la recta de referencia y su perpendicular. Este nuevo sistema de coordenadas permite obtener directamente las distancias de las zonas afectadas de las líneas, a la recta de referencia (coordenada x).

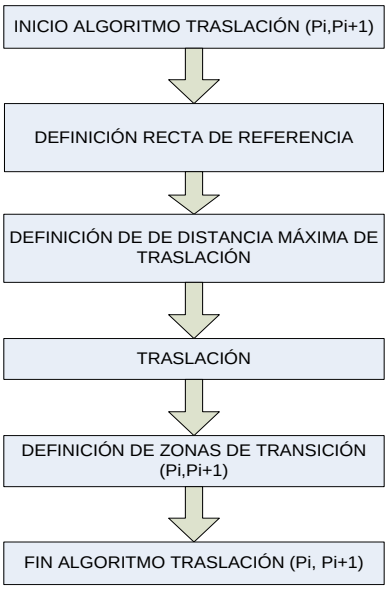


Figura 4.38. Algoritmo de traslación del corte.

El siguiente paso será evaluar la distancia máxima con signo de cada una de las trazas a la recta de referencia. La suma de estas distancias será la *distancia máxima de traslación*. Una vez que se ha redefinido la zona entre los puntos de corte se deshace el cambio de base obteniendo como resultado la nueva nube de puntos sin cortes.

A continuación se definirán unas zonas denominadas *zonas de transición* (Figura 4.39), colocadas en los extremos de la *zona de corte*. Estas zonas van a garantizar que el resultado sea contínuo y no presente cambios bruscos de curvatura. Heurísticamente, siguiendo criterios perceptivos y teniendo en cuenta la normativa circulatoria, se ha comprobado que un tamaño óptimo para estas zonas es la mitad de la distancia máxima de traslación. Sin embargo características particulares del corte o la posible existencia de puntos con posiciones fijas (puntos de inicio y fin de zonas paralelas, puntos de inicio o de fin de zonas restringidas, etc) puede obligar el cambio de estas magnitudes, por lo que dicha variable puede ser modificada manualmente en caso necesario.

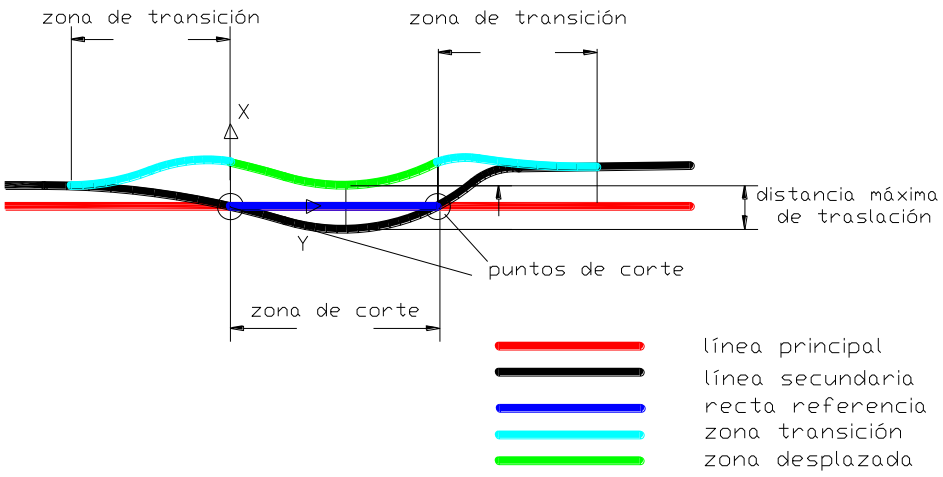


Figura 4.39. Proceso de resolución del corte.

Estas zonas anterior y posterior se van a unir a la zona trasladada mediante una consolidación y suavizado, utilizando como pivote los puntos extremos de cada intervalo (Figura 4.40).

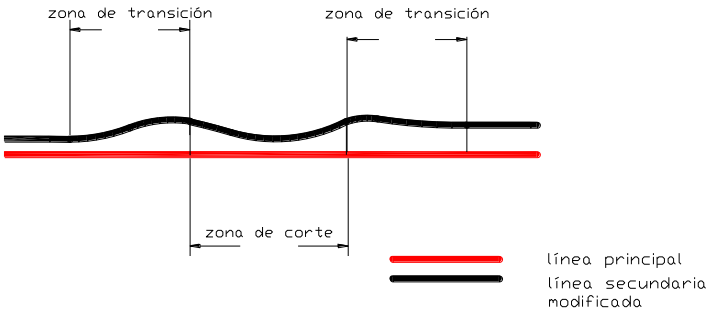


Figura 4.40. Líneas tras la resolución del corte.

En el momento en que se dispone de la nueva definición geométrica de la zona de corte, se comprueba si su longitud es admisible. En el caso de no serlo se tendrá que aplicar el algoritmo de aplanamiento (Figura 4.41) y se volverá a evaluar la deformación longitudinal sufrida.

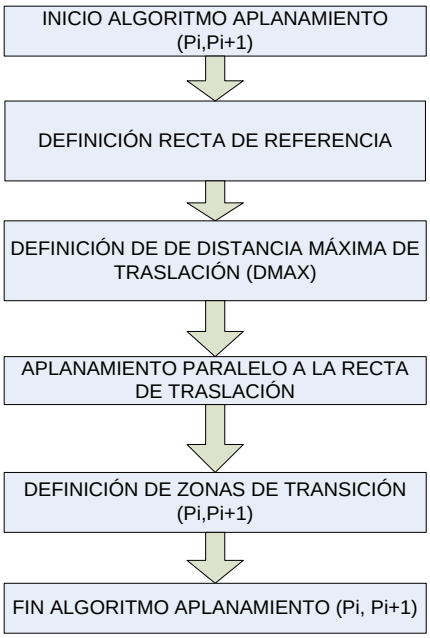


Figura 4.41. Algoritmo de aplanamiento del corte.

El no cumplimiento de la tolerancia longitudinal admisible en este caso, significa que el error introducido por los planos de señalización es demasiado complejo para poder ser resuelto automáticamente. Los planos habrán de ser revisados y las tablas de configuración y/o la línea, habrán de ser editadas manualmente.

4.5 CONSTRUCCIÓN DE LA RED DE TRAYECTORIAS.

A la hora de afrontar la generación (conectividades, consolidación y suavizado) de una determinada trayectoria perteneciente a un entorno ferroviario, urbano o interurbano, es necesario

tener en cuenta las relaciones topológicas y geométricas existentes con las restantes trayectorias, además de las restricciones impuestas por la normativa circulatoria, módulo de conducción y proyecto. Todo esto ha motivado el desarrollo, dentro de la Tecnología Modular, de un apartado centrado en la generación de la red de conexiones con los requisitos y los datos de partida manejados por los simuladores desarrollados por el CITEF.

A continuación, se detalla el proceso constructivo que la presente Tesis ha desarrollado para generar las redes de trayectorias que son empleadas en los simuladores de conducción terrestre del CITEF. Este proceso se sustenta en el empleo del Módulo de Generación Topológica y del Módulo de Ajuste Geométrico definidos en los apartados anteriores.

4.5.1 ENTORNOS FERROVIARIOS.

La metodología creada por esta Tesis para definir geométricamente un entorno ferroviario comienza con el establecimiento de una jerarquía de prioridades constructivas, basada en la taxonomía de líneas derivada de las definiciones topológicas establecidas por esta Tesis para la creación de entornos ferroviarios.

En primer lugar se generarán las líneas ficticias, cuya funcionalidad es suministrar la descripción geométrica de las restantes líneas mediante relaciones de paralelismo. La metodología a seguir para ello dependerá de la información disponible:

- En el caso de disponer de la definición de radios y pendientes, se empleará como elemento base el biarco. Si además se dispone de información topográfica, la línea se someterá a un proceso de consolidación y suavizado. Este suavizado se realiza ajustándose a los parámetros de generación del proyecto (radios mínimos, tolerancia de deformación longitudinal máxima y mínima., etc)
- En el caso de disponer únicamente de información topográfica, la nube de puntos representativa del trazado se someterá a un proceso de suavizado respetando los parámetros constructivos comentados previamente.

Una vez generada la planta de las líneas ficticias se generarán el resto de líneas según su prioridad constructiva. Esta prioridad viene motivada por las interdependencias existentes entre ellas, debido al comienzo o finalización de unas en otras, lo cuál exige que la línea de la que se nace o en la que se finaliza, esté definida previamente para garantizar la continuidad espacial y tangencial. Según esto se establecen las siguientes prioridades:

- Líneas normales: prioridad constructiva 1.
- Líneas que comienza en líneas (Figura 4.42): prioridad constructiva 2. Esto es debido a que para su construcción es necesario que las líneas con prioridad constructiva 1 estén ya generadas, ya que el intervalo de conexión con las mismas exige una continuidad espacial y tangencial.
- Líneas que terminan en líneas (Figura 4.43): prioridad constructiva 2, por las mismas razones que las anteriores.
- Líneas que comienzan y terminan en líneas: prioridad constructiva 2, por las mismas razones que las dos anteriores.
- Agujas: prioridad constructiva 3. Se trata de geometrías que comienzan y terminan en líneas de prioridad constructiva 1 y 2, así que con el fin de garantizar la continuidad espacial y tangencial en el intervalo de conexión con las mismas, es necesario que su definición se realice en último lugar.

Figura 4.42. Ejemplo de línea que comienza en línea (túnel de enlace).

Figura 4.43. Ejemplo de línea que termina en otra línea (LEL).

Además será necesario tener en cuenta su posición respecto de la línea ficticia (posición 0), siendo mayor la prioridad cuanto mayor es la proximidad a ésta. En el caso de existir líneas cuya prioridad constructiva infrinja las reglas anteriores, se definirá explícitamente dicha prioridad.

La definición geométrica de cada línea se realizará mediante la aplicación de los algoritmos de consolidación, suavizado y gestión de cortes descritos en el Módulo de Ajuste Geométrico. Para ello se tendrá en cuenta que la definición geométrica del entorno se realiza en base a dos tipos de zonas:

- **Zonas estáticas o predefinidas**: aquéllas cuya definición geométrica suministrada por las fuentes de información de partida no puede ser alterada.
- **Zonas flexibles o no predefinidas**: aquéllas que podrán ser modificadas con el fin de absorber los errores asociados a las fuentes de información de partida.

La presencia de zonas predefinidas y vinculaciones geométricas entre líneas fuerzan a que la consolidación se realice por intervalos, siendo los intervalos de independencia no predefinidos los únicos que habrán de consolidarse, ya que los restantes tienen su geometría restringida (ya sea por poseer relaciones de paralelismo con otras líneas, ya sea por circunstancias específicas, como por ejemplo, la definición geométrica de una estación que no puede ser alterada, que impidan modificar su definición geométrica inicial).

Dentro de cada zona a consolidar, el algoritmo calculará la distancia entre la nube de puntos inicial y la topográfica, e irá consolidando únicamente los puntos que superen una determinada tolerancia de error. Dicha tolerancia dependerá del proyecto en cuestión. El objetivo de no consolidar todos los puntos del intervalo es disminuir los errores tangenciales introducidos en los extremos del mismo por el proceso de consolidación, que si bien son eliminados posteriormente con el algoritmo de suavizado, originan excesivos cambios de radio que pueden provocar un aspecto visual incorrecto debido a un excesivo zigzagueo.

Una vez consolidado un intervalo se suaviza y se prosigue con la consolidación del siguiente intervalo de independencia que exceda la tolerancia de error mencionada.

A la hora de afrontar el suavizado se distinguen los siguientes tipos de intervalos:

- **Intervalos independientes**: Se suavizan siguiendo el procedimiento general de suavizado a gran escala (apartado 4.4.2.2). Se aplica a todos aquellos intervalos de trayectoria que o bien no poseen ningún tipo de restricción topológica y han sufrido un proceso de consolidación, o bien sus datos han sido obtenidos directamente por un proceso de lectura automática del plano topográfico.

- **Conexión:** Se define una conexión como un biarco simple generado a partir de la interpolación de Hermite de los datos suministrados por dos trazados situados en sus extremos, que pueden pertenecer a la misma trayectoria o a trayectorias diferentes. Ejemplos del primer caso se encuentran en las entradas y salidas a estaciones, en las líneas que comienzan en líneas (LBL), líneas que terminan en líneas (LEL) y líneas que comienzan y terminan en líneas (LB&EL). Las siguientes muestran algunos ejemplos de estas conexiones (Figura 4.44).

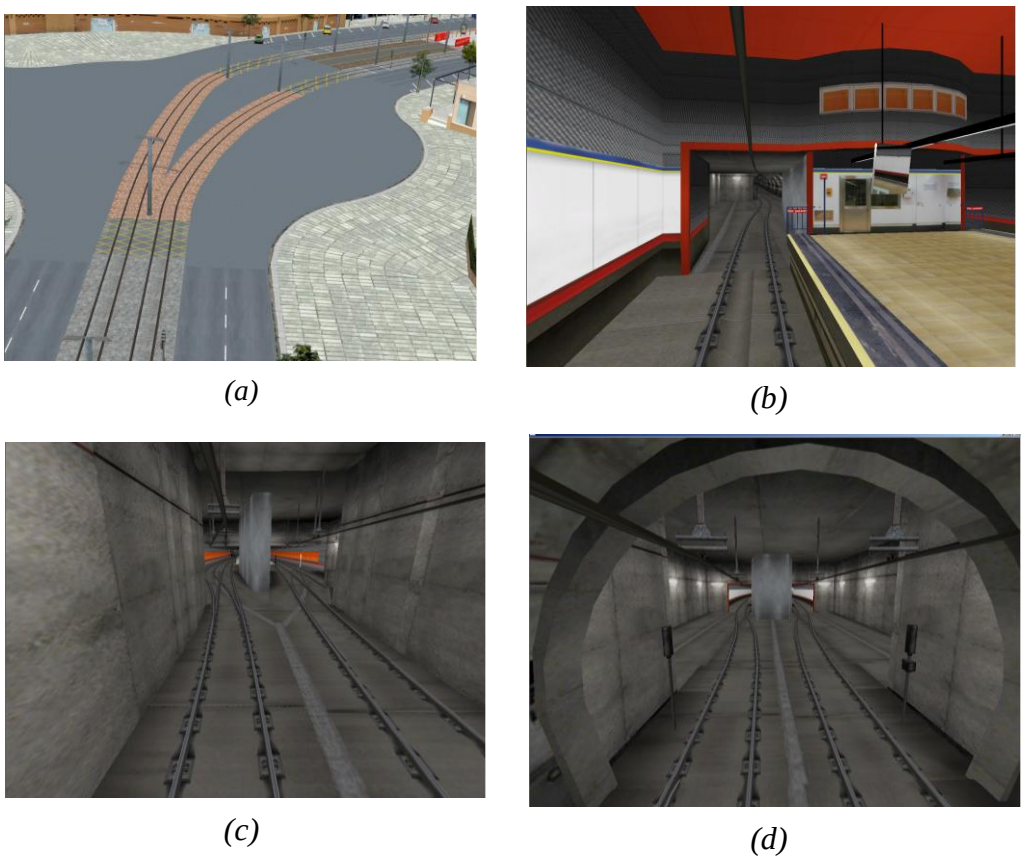


Figura 4.44. Ejemplos de intervalos de tipo conexión: (a) unión de dos zonas con diferente distancia de paralelismo;(b) salida de una estación; (c) línea que comienza en línea, aguja y conexión en entrada a estació;.(d) entrada a estación.

Un ejemplo del segundo caso, conexión entre dos trayectorias diferentes, lo constituyen las agujas. Para éstas resulta fundamental disponer de este algoritmo que genere su geometría automáticamente, ya que los planos no suelen suministrar información geométrica de las mismas (Figura 4.45).

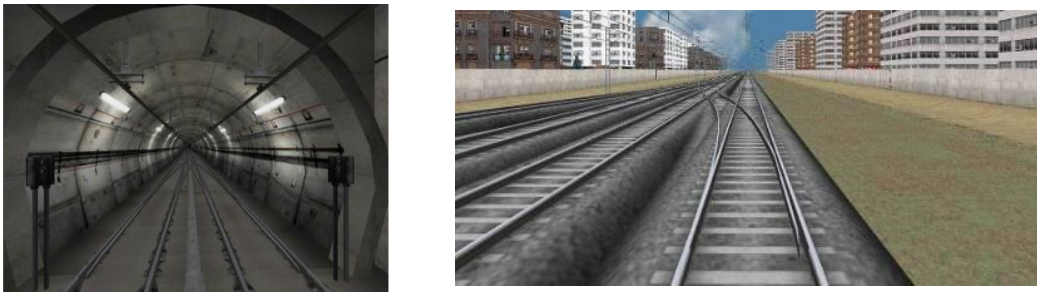


Figura 4.45. Ejemplos de generación automática de aguja

- **Clotoides:** Las clotoides son curvas de transición que se encargan de dotar al trazado de una continuidad en curvatura. Se caracterizan por la variación lineal de su curvatura con la longitud. El suavizado de estos intervalos se realiza mediante una curva biarco constituida por n+1 arcos. Con n>=1 y estando los arcos numerados de 0 a n, sea S la longitud de la clotoide. Entonces los arcos 0 y n tendrán longitud S/(2n) mientras los arcos desde 1 hasta n-1 tendrán longitud S/n. Las curvaturas de los n+1 arcos seguirán una progresión geométrica:

  $k_0=k_A$, $k_j=k_0+jh$, j=0,1…,n y $k_n=k_B$, donde $h=\frac{k_B-k_A}{n}$, siendo A el punto de inicio de la clotoide con curvatura k_A y B el punto de fin con curvatura k_B.

- **Acuerdo vertical**: Se define el acuerdo vertical como la curva de transición que enlaza dos pendientes. En función de la velocidad a la que se permita viajar por un determinado trazado, este acuerdo será un arco de circunferencia, si la velocidad es moderada o una parábola (que se considerará siempre simétrica) para velocidades elevadas.
  En el caso de que el acuerdo se defina como un arco, el suavizado de estos intervalos mediante una curva biarco será inmediato. Por cada par de pendientes suministrada se genera el biarco constituido por las entidades recta-arco-recta. Las rectas tendrán como pendiente la suministrada y como punto de inicio el definido por el pk absoluto de inicio del intervalo. El arco de radio R, será el tangente a ambas.
  En el caso de que el acuerdo sea una parábola, se suministra la información de la distancia entre puntos de tangencia (parámetro Lv mostrado en Figura 4.46).La única diferencia respecto del caso anterior consiste en la sustitución del arco de acuerdo por un biarco simple. El cálculo del biarco simple responderá a la interpolación de Hermite de los puntos de tangencia calculados a partir del parámetro Lv.

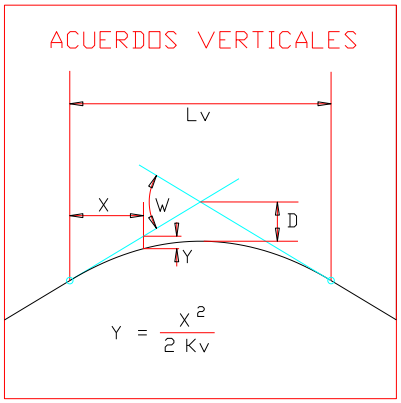


Figura 4.46. Geometría del acuerdo vertical.

Una vez concluida la generación de la planta de la línea se revisarán posibles cortes con otras líneas y en caso de existir, se resolverán aplicando los algoritmos de gestión de cortes descritos previamente. Resueltos los cortes y concluida la definición geométrica de la planta se llevará a cabo la consolidación.

Todo el proceso constructivo descrito para la generación de la planta se aplica a continuación al perfil. El tratamiento es idéntico vasta sustituir la coordenada X por la coordenada PK (punto kilométrico) y la coordenada Y por la coordenada Z (2.5D). La única diferencia estriba en los parámetros de suavizado y las tolerancias admisibles en uno y otro caso. Finalmente se procederá al cálculo del peralte teniendo en cuenta la normativa que rija dicho trazado.

4.5.2 ENTORNOS URBANOS.

La metodología creada por esta Tesis para definir geométricamente un entorno urbano comienza con la generación de las curvas constructivas. Mediante la aplicación de los algoritmos de consolidación, suavizado y resolución de cortes, se garantizará que cada curva constructiva pasa por todos sus nodos con continuidad tangencial. Al igual que en un entorno ferroviario, la metodología a seguir dependerá de la información disponible:

- En el caso de disponer de la definición de radios y pendientes, se empleará como elemento base el biarco. En el caso de un entorno urbano, los nodos son siempre puntos de obligada consolidación.

- En el caso de disponer únicamente de información topográfica, la nube de puntos representativa del trazado se someterá a un proceso de suavizado respetando los parámetros constructivos comentados previamente.

A continuación se dividirá cada curva constructiva en tantos fragmentos (nuevas curvas constructivas) como tramos entre nodos existan. El motivo por el que algunas curvas constructivas no se definen desde el principio únicamente por dos nodos, es garantizar de una manera más sencilla mediante el biarco proporcionado por el algoritmo de suavizado, la continuidad tangencial en todos ellos.

Posteriormente se definirá el contorno de las intersecciones. Con el fin de facilitar su construcción, los diferentes tipos de nodos se han agrupado en función de sus similitudes geométricas en las siguientes categorías que en el apartado siguiente serán descritas en detalle:

- Nodos cruce y rotonda: abarcan los nodos de cruce y rotonda.

- Nodos T: abarcan los nodos de incorporación y salida de carril.

- Nodos variación: abarcan los nodos de aumento y de disminución del número de carriles.

El contorno de las intersecciones servirá para delimitar los carriles ficticios: porción de curva constructiva comprendida entre los contornos de sus intersecciones extremas. Definidos así los carriles ficticios se procederá a la generación de los restantes carriles siguiendo para cada ramal, el mismo proceso constructivo que el llevado a cabo para definir un LDL en un entorno ferroviario.

4.5.3 CONSTRUCCIÓN DE LAS INTERSECCIONES.

NODOS CRUCE Y ROTONDA.

La descripción geométrica del contorno de la intersección que define un nodo cruce o rotonda, comienza con la creación de la lista de curvas constructivas que confluyen en la misma. En esta lista, las curvas constructivas aparecen ordenadas según el ángulo creciente que forman con el eje x (sentido positivo el contrario a las agujas del reloj). Los ángulos serán los formados por los vectores unitarios de las curvas en sentido de salida del nodo, no se tiene en cuenta por tanto el sentido intrínseco a la curva constructiva.

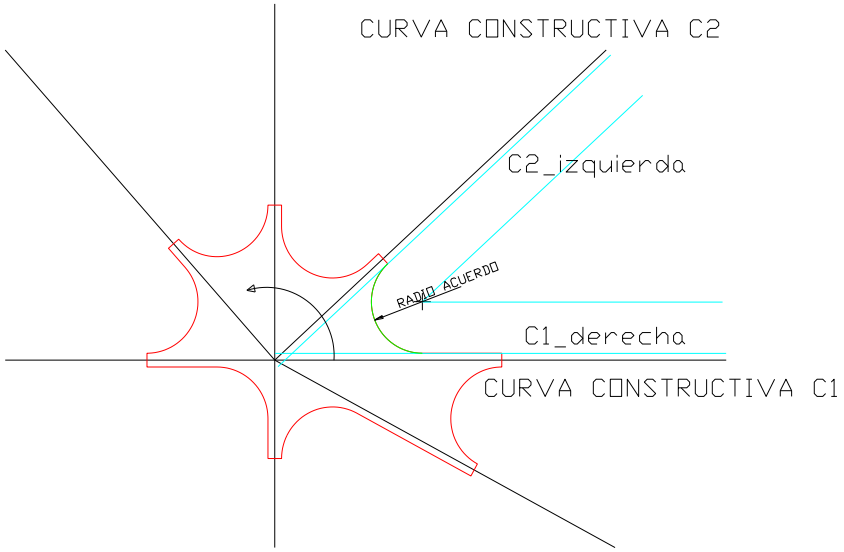


Figura 4.47. Construcción geométrica de un Nodo Cruce.

A continuación para cada par de curvas constructivas, C1 y C2 (Figura 4.47), se calcula su geometría de acuerdo: radio tangente a los carriles más extremos de ambos ramales (C1_derecha y C2_ izquierda). En el caso de un nodo de tipo Rotonda (Figura 4.49), se habrá de tener en cuenta si el contorno tendrá que absorber o no a la curva que define el exterior de la misma. En este último caso, será necesario el cálculo de dos radios de acuerdo, tangentes a los carriles más extremos de uno y otro ramal y a la curva que define la geometría más externa de la rotonda.

Figura 4.48. Nodos de tipo Rotonda.

Una vez definidas las dos zonas de acuerdo, derecha e izquierda, de cada curva constructiva se procede a cerrar el contorno de la intersección. Este contorno servirá para delimitar los comienzos o fines de ramal. Para ello se evalúa para cada curva constructiva la distancia entre el punto de tangencia del radio de acuerdo derecho e izquierdo al nodo de la intersección. El punto más distante, será el que se emplee para cerrar el contorno. El objetivo de esta selección es permitir una construcción modular de los carriles.

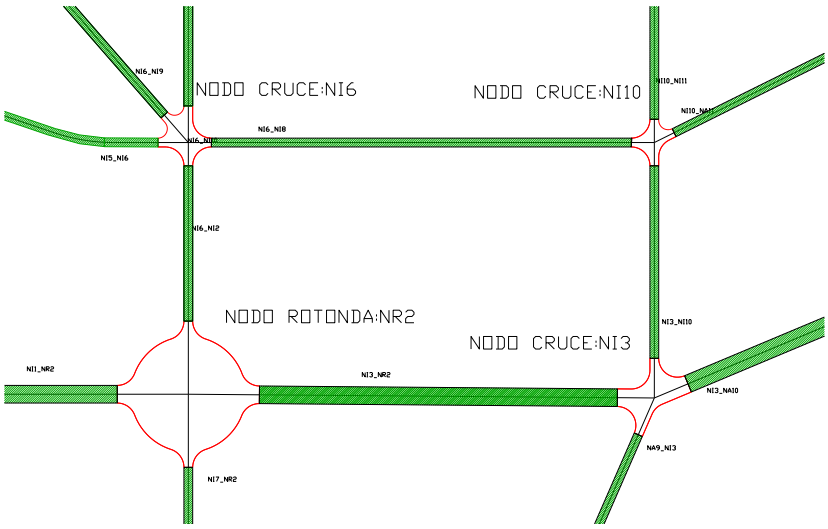


Figura 4.49. Ejemplos de nodo Cruce y nodos Rotunda.

Las siguientes figuras muestran los módulos y la malla que definen los carriles y la intersección respectivamente, que confluyen en un nodo de tipo Cruce.

(a) (b) (c)

Figura 4.50. Nodo de tipo Cruce: malla y módulos: (a) malla intersección; (b) módulo de carril; (c) módulo de marca vial.

NODOS T.

Con este tipo de intersecciones se representan las salidas e incorporaciones de carril. Se caracterizan por la existencia de una curva constructiva a la que se designa como principal y otra curva que converge/diverge en ella y que se considera secundaria. La curva constructiva principal deberá estar constituida por al menos tres nodos, siendo el nodo intermedio, el nodo T. De esta manera los algoritmos de consolidación y suavizado garantizan la continuidad tangencial en dicho nodo. Los ramales a los que de lugar la curva constructiva principal, deberán tener el mismo número de carriles antes y después del nodo T.

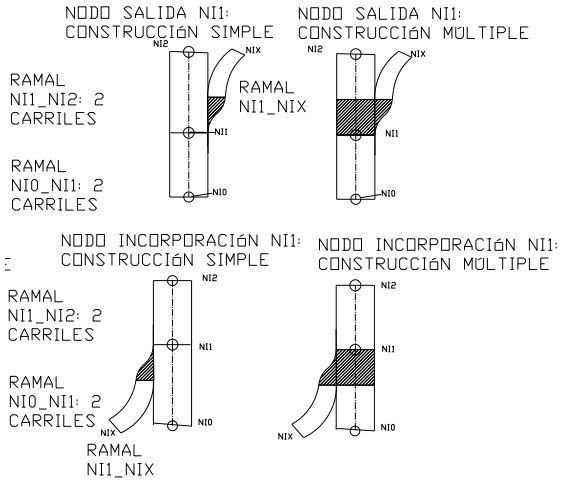


Figura 4.51. Nodos T.

Con el fin de potenciar las ventajas de la Tecnología Modular se han desarrollado dos posibles construcciones para este tipo de intersecciones:

- **Construcción simple**: La ventaja de este método es conservar la definición modular de carriles en la dirección principal. Esto facilitará la inserción de entornos y marcas viales, además de las ventajas propias de estos módulos. Sin embargo, con el fin de delimitar correctamente el área de intersección y las rutas asociadas a la misma, su introducción implica la definición de unos *nodos circulatorios* (Figura 4.52) desacoplados de los nodos intersección.

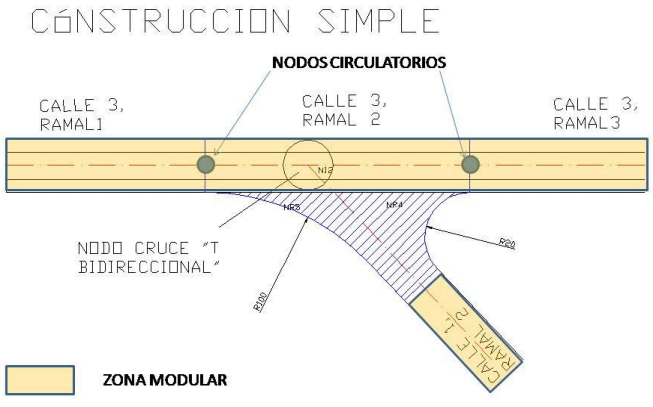


Figura 4.52. Construcción simple de nodos T.

- **Construcción múltiple**: En este caso toda la intersección (zona rallada en azul en la Figura 4.53), se malla. Pese a no aprovecharse la zona modular, conserva la metodología aplicada hasta el momento, facilitando el cálculo de rutas necesarias para el modelo de tráfico.

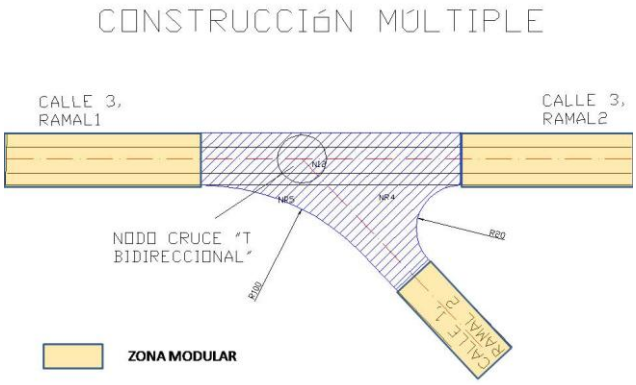


Figura 4.53. Construcción múltiple de nodos T.

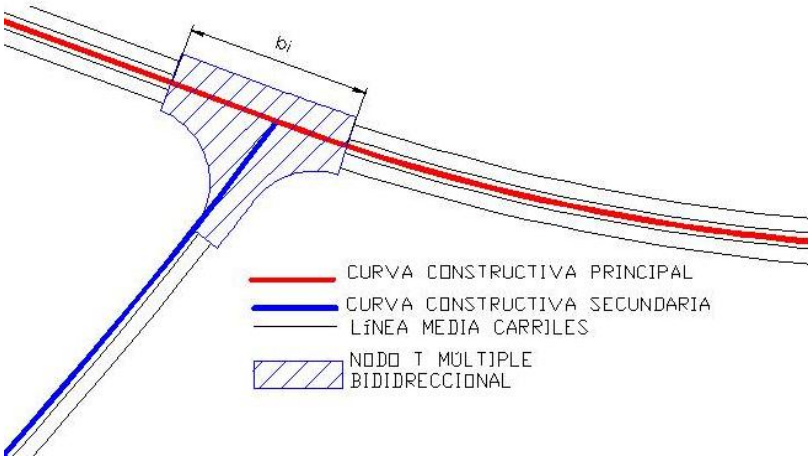
A su vez, se distinguen dos tipos de nodos T: bidireccional y unidireccional.

NODO T BIDIRECCIONAL

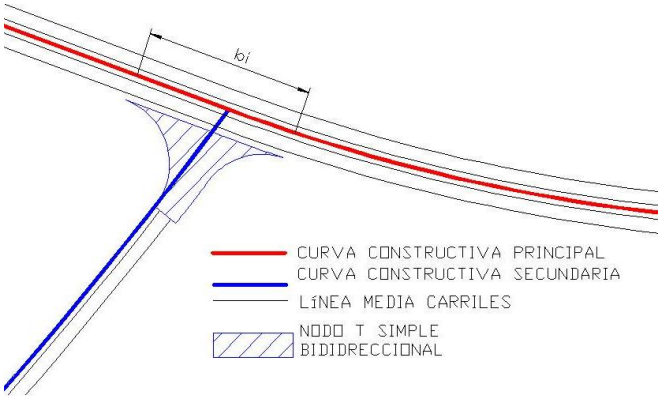
El nodo bidireccional va a permitir la entrada y salida hacia el ramal secundario.

Figura 4.54. Ejemplos de nodos T bidireccionales.

En primer lugar se calcularán las zonas de acuerdo entre la curva principal y la secundaria. A continuación para cerrar la intersección, se calcula el tramo de curva constructiva principal (biarco intermedio bi en la 0), orientado en sentido contrario al de las agujas del reloj. Una vez que se tiene el biarco intermedio, se calcula la paralela que define el contorno de la intersección que limita con el ramal principal. En el caso de una *construcción simple* esta paralela se calculará hacia el lado de convergencia/divergencia de la curva secundaria y en el caso de la *construcción múltiple* hacia el lado contrario.



(a)



(b)

Figura 4.55. Nodo T Bidireccional: (a) construcción simple; (b) construcción múltiple.

NODO T UNIDIRECCIONAL

Como su nombre indica, este tipo de intersección solo va a permitir la entrada ó salida hacia el ramal secundario. Por este motivo solo llevará acuerdo el lado de la curva secundaria que forme un mayor ángulo con la curva constructiva principal.

La metodología a seguir es la misma que en el caso de los nodos T bidireccionales considerando que en uno de los lados no va a existir acuerdo, por lo que dicha zona se va a calcular por intersección entre los contornos exteriores de ambos ramales.

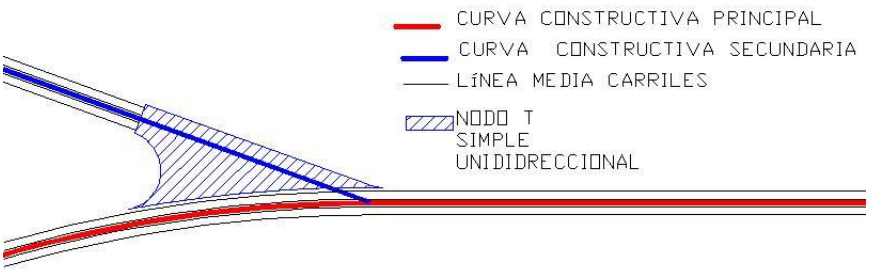


Figura 4.56. Nodo T Unidireccional simple.

NODOS VARIACIÓN.

Mediante los nodos Variación se representan las zonas donde tienen lugar un aumento o disminución del número de carriles. En estas intersecciones va a existir una única curva constructiva, lo que garantizará, mediante algoritmos de consolidación y suavizado, la continuidad tangencial en el nodo Variación (Figura 4.57).

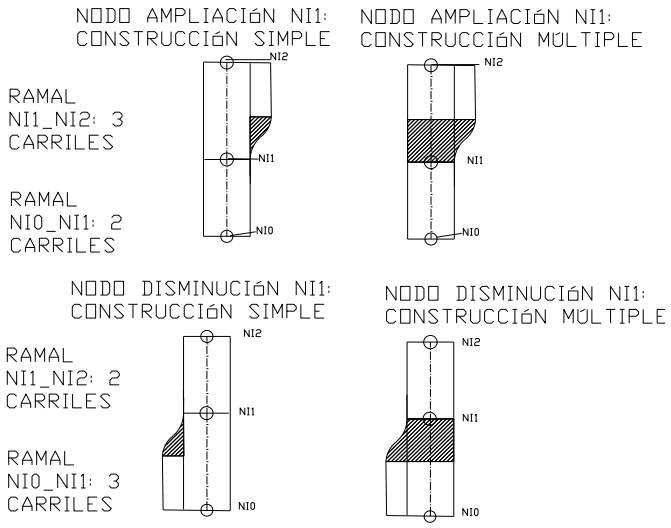


Figura 4.57. Construcción nodos Variación de carril.

La construcción es similar al caso de un nodo T (*construcción simple y múltiple*) salvo en la curva de transición que cierra el contorno en la zona de aparición/ desaparición del carril. Dicha curva es un biarco constituido por dos arcos, que responde a la interpolación de Hermite de los siguientes datos (Figura 4.58):

- Punto Pi, con tangente ti: punto de intersección de la perpendicular a la curva constructiva por el nodo Variación, con el contorno más externo del ramal. La tangente ti, será la dirección de la tangente a la curva constructiva en ese punto.
- Punto Pf, con tangente tf: se calcula el punto de la curva constructiva N', que dista D del nodo Variación. D es un parámetro de entrada. Se calcula el punto de intersección de la perpendicular a la curva constructiva por N' con el contorno más externo del ramal, Pf. La tangente tf, será la dirección de la tangente a la curva constructiva en N'.

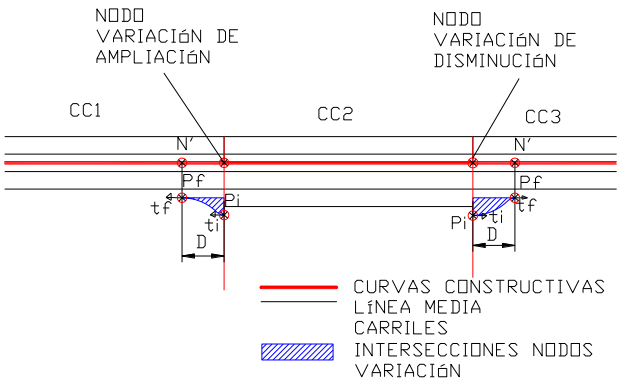


Figura 4.58. Construcción de nodos Variación de carril.

4.6 CONSTRUCCIÓN DEL ENTORNO CIRCUNDANTE.

La construcción del entorno circundante consistirá en la definición de las familias modulares a insertar en cada punto kilométrico del mismo, lo que exigirá la creación y almacenamiento previo de los perfiles transversales y texturas que las definen. En aquéllas zonas donde la construcción modular no sea posible se crearán las mallas necesarias.

En la presente Tesis el almacenamiento de toda la información geométrica relacionada con el entorno es de tipo vectorial, simplificando así al máximo el espacio de almacenamiento. No será necesario almacenar la malla (TIN) de estos elementos sino únicamente sus perfiles transversales, contornos y texturas y en el momento de su representación virtual podrán generarse automáticamente a partir de las herramientas desarrolladas por la Tecnología Modular. La única excepción serán los elementos singulares creados ad hoc por el sector infografista (estaciones, edificios emblemáticos, etc)

En la extracción y generación de perfiles transversales se distinguen dos casos: elementos situados sobre el terreno y el propio terreno.

En el primer caso los perfiles serán extraídos de los planos suministrados o en su defecto de las imágenes que se dispongan de la zona en cuestión, como es el caso de los túneles en una red metropolitana. Su almacenamiento pasará por la introducción de dicha información en la base de datos de modelos con la ayuda de las herramientas generadas por la Tecnología Modular [292].

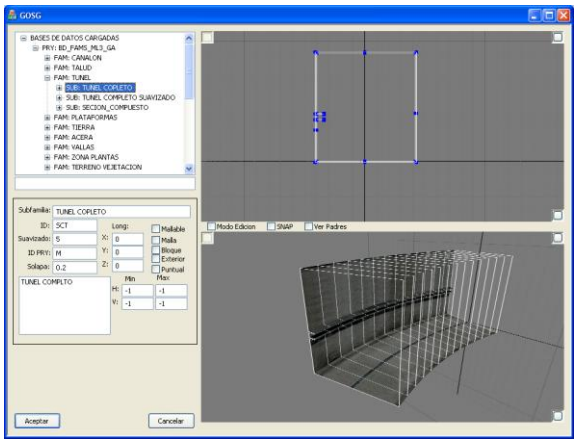


Figura 4.59. Introducción del perfil transversal de un túnel.

En el caso particular de la generación del terreno se trata de sintetizar su definición en base a un conjunto de perfiles transversales que estarán agrupados según su posición lateral respecto de la línea directriz que rige su posicionamiento (derecha, izquierda o centrado) y según la distancia transversal a la misma, dando lugar a diferentes niveles de proximidad. De esta manera se consigue una mayor versatilidad constructiva y optimización gráfica, ya que a medida que nos separamos de la línea directriz el nivel de detalle es menor lo que a su vez permite el empleo de niveles de detalle discretos variantes con el punto de vista. Las distancias de corte son elegidas por procedimientos heurísticos para cada tipo de entorno. La orografía del terreno y posibles obras civiles (túneles, pasos a nivel, etc) serán cruciales en dicha decisión.

El proceso para la determinación de la definición geométrica y el posicionamiento de los perfiles transversales que definen el terreno, variará en función de si se trata de un entorno real o

[292] Anexo 3.

ficticio. En ambos casos la generación del entorno exigirá la definición previa de la geometría de las líneas directrices que actuarán como guías para el posicionamiento modular. A continuación se describen ambos casos.

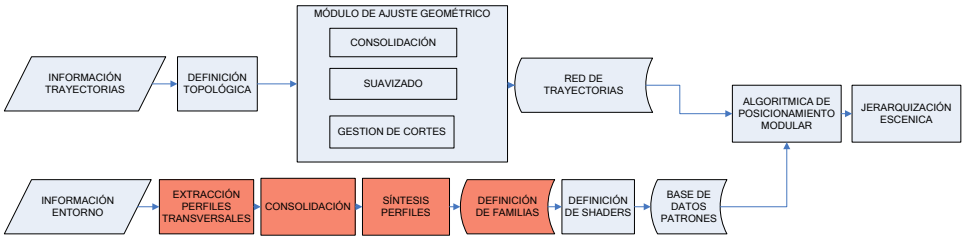


Figura 4.60. Esquema de funcionamiento de la Tecnología Modular.

4.6.1 SÍNTESIS DE PERFILES TRANSVERSALES EN LA CONSTRUCCIÓN DE ENTORNOS REALES.

El proceso de síntesis del entorno circundante en un conjunto de perfiles transversales representativos del mismo implica la resolución de las siguientes fases:

1. EXTRACCIÓN DE LOS PERFILES TRANSVERSALES.

   Si se tiene como información de partida un plano topográfico con información de isolineas que definen la altitud del terreno y las trayectorias circulatorias, para cada línea directriz [293] se hace un barrido de perfiles: se traza a cada metro un corte por un plano perpendicular a la línea directriz, determinando su intersección con las distintas isolineas y generando así los perfiles transversales

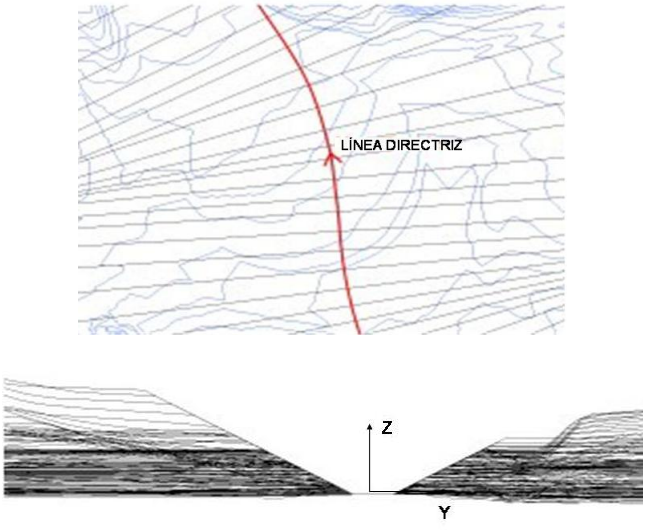


Figura 4.61. Generación de perfiles por corte con las isolineas.

[293] No todas las trayectorias circulatorias deben actuar necesariamente como líneas directrices. La elección de estas trayectorias dependerá del entorno a representar. La existencia de trayectorias no paralelas podría dar lugar a la aparición de huecos que fuesen necesario cubrir con mallas ad hoc según se comentó en el apartado 3.5 del Capítulo 3.

En el caso de disponer de un Modelo digital de Elevaciones (MDE) el proceso será idéntico, esta vez hallando las intersecciones con las celdas del MDE. Generalmente este último proceso llevará a la obtención de soluciones menos precisas, ya que la precisión del MDE no suele ser la misma que la del mapa vectorial que define la trayectoria.

2. CONSOLIDACIÓN DE LOS PERFILES TRANSVERSALES.

Toda línea ferroviaria irá acompañada de unos planos que definan el trazado en planta y perfil de la misma (Figura 4.62). La mayor precisión y generalmente actualización de estos planos respecto de los topográficos que definen el entorno exige la consolidación de la zona adyacente a la línea directriz [294].

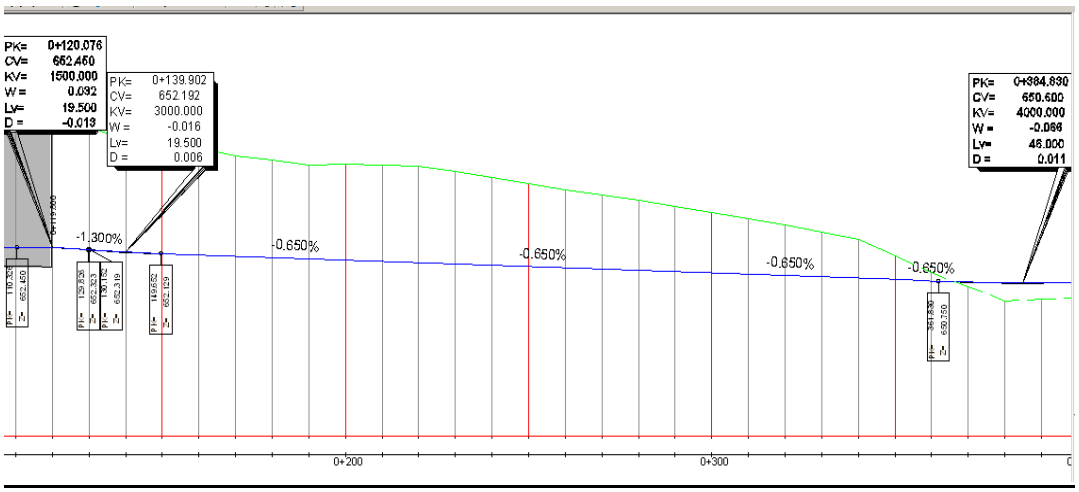


Figura 4.62. Ejemplo de plano con información de gradientes, desmontes y terraplenes empleado para consolidar.

Una vez consolidados estos perfiles serán subdivididos en diferentes niveles de proximidad con el fin de favorecer la construcción modular del entorno [295]: la fragmentación transversal aumentará la flexibilidad constructiva del entorno y la adecuación del nivel de detalle en función de la distancia transversal a la línea directriz.

3. SÍNTETIS DE PERFILES.

Con el fin de definir las familias de terreno es necesario sintetizar el conjunto de perfiles transversales obtenido en un conjunto de perfiles representativo del mismo. Para ello, para cada línea directriz y nivel de proximidad, se aproximan todas las nubes de puntos que definen los perfiles transversales mediante splines (Figura 4.63 (a)).

[294] Es frecuente que existan discrepancias entre la cota z asociada a las líneas directrices, que viene definida a través de sus gradientes y la representada en los mapas de isolineas y MDEs (éstos no suelen contener la información de las obras civiles llevadas a cabo). Junto a los planos con información de gradientes suele adjuntarse información sobre desmontes y terraplenes, que es la fuente de información que se emplea para consolidar la información de cota Z asociada al primer nivel de proximidad, el colindante con la línea directriz.

[295] En los entornos virtuales generados para los simuladores de conducción del CITEF, se ha determinado que el número de niveles de corte transversal óptimo se encuentra en tres. Los resultados de las pruebas de rendimiento específicas para un determinado entorno, como se verá en el Capítulo 5, pueden llevar a la modificación de este número.

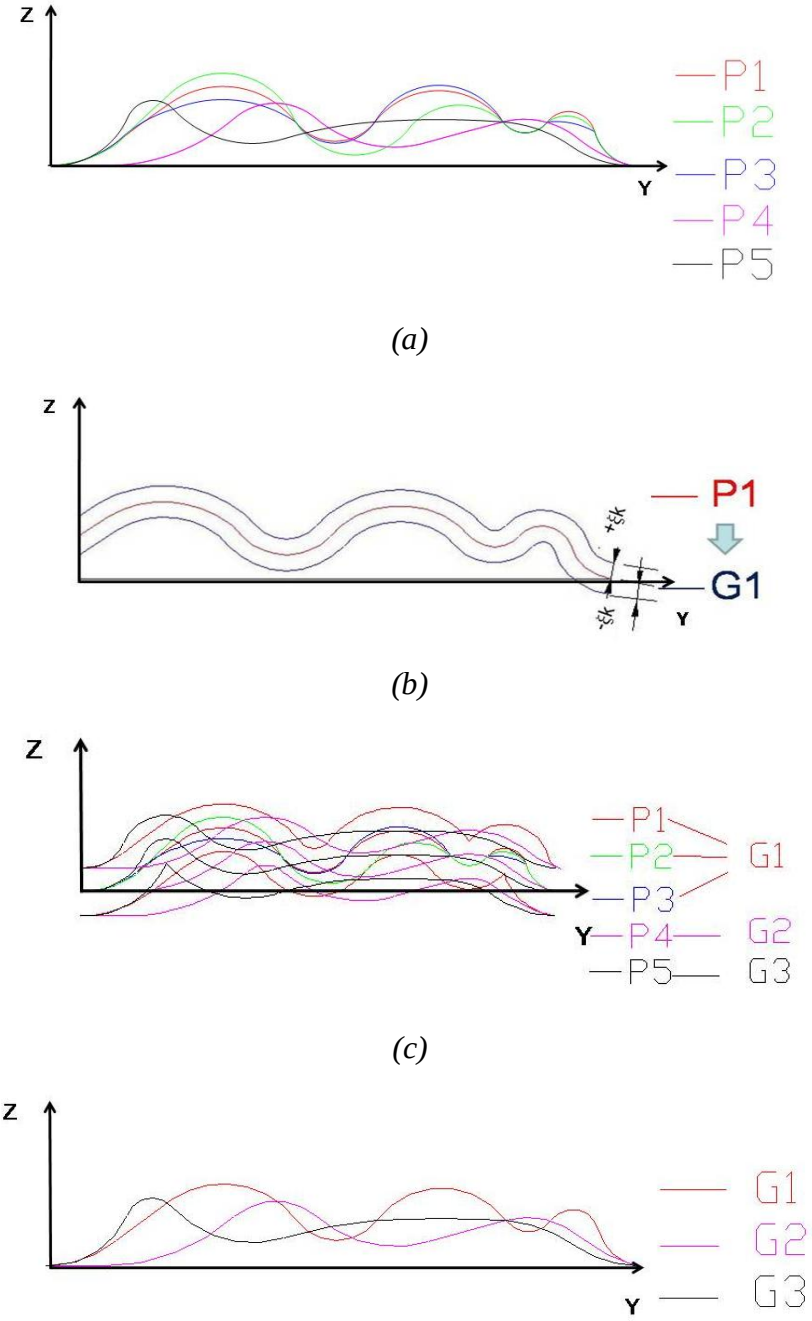


Figura 4.63. Ejemplo de síntesis de perfiles transversales: (a) perfiles transversales definidos mediante splines; (b) Definición del primer grupo de perfiles G1; (c) Definición del segundo grupo de perfiles G2; (d) Definición del tercer grupo de perfiles G3.

Se toma el primer perfil y se define con él el primer grupo de perfiles.Todos los perfiles que se encuentren dentro de un offset de ξk, donde ξk depende del nivel de proximidad, pertenecerán al grupo 1 (G1, Figura 4.63 (b)). El primer perfil que no pertenezca a este offset, pasa a constituir la semilla de un nuevo grupo 2 (G2, Figura 4.63 (c)), definiéndose nuevamente su offset como ξk. El siguiente perfil, si no pertenece a ninguno de los grupos anteriores definirá su propio grupo y así sucesivamente (G3, Figura 4.63 (d)).

4. DEFINICIÓN DE LA SECUENCIA LONGITUDINAL DE PERFILES

Hasta el momento se dispone, para cada línea directriz y nivel de proximidad, de una secuencia de perfiles transversales optimizados definidos a cada metro. Sin embargo, se requiere una definición del entorno que permita la creación del mismo a partir de la concatenación de módulos de tamaño la longitud básica, Lb. Si sin más, se toman los perfiles existentes cada múltiplo de Lb podría ocurrir que se omitiesen perfiles que fuesen decisivos para conseguir un grado de similitud con la realidad adecuado. Para ello se establece la secuencia longitudinal de perfiles del entorno, que será un listado de intervalos que contienen los puntos de cambio de perfil transversal. Estos intervalos vendrán caracterizados por su pk de inicio, pk de fin, lateralidad, nivel de proximidad y la poligonal que define su perfil. La siguiente figura muestra un ejemplo de perfiles transversales generados a partir del editor de perfiles [296] desarrollado para el CITEF y su empleo en el simulador de Metro Ligero de Madrid.

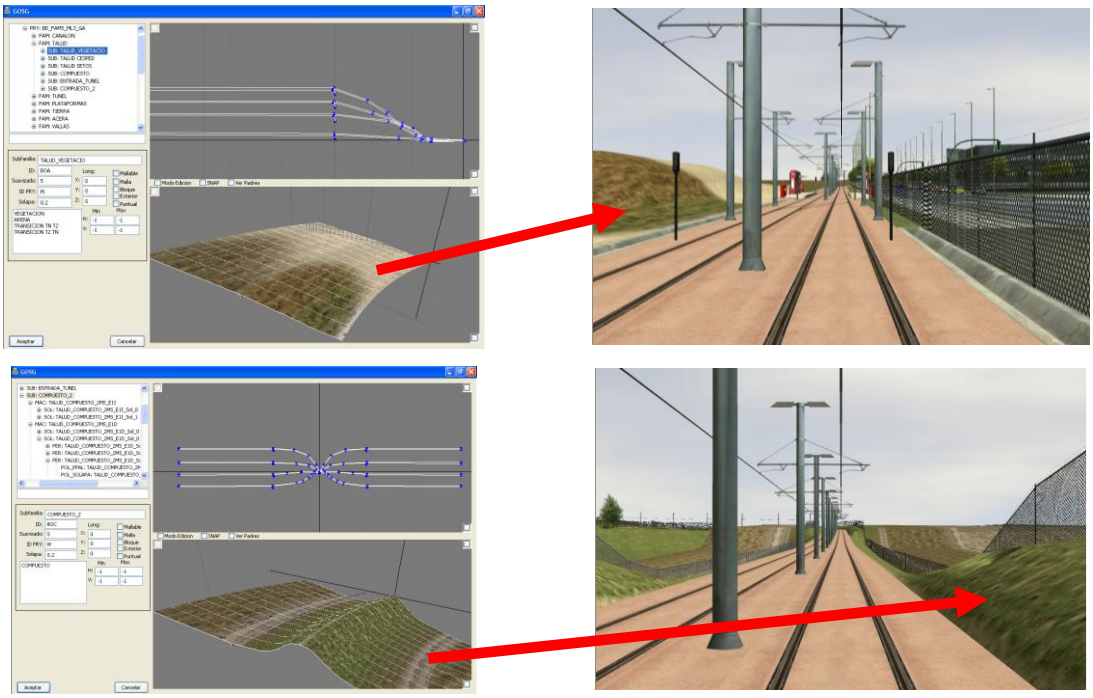


Figura 4.64. Ejemplos de definición de perfiles transversales de terreno y su empleo en el simulador de Metro Ligero de Madrid.

5. CODIFICACIÓN DE TALUDES.

Una vez definidos los intervalos longitudinales del entorno y almacenadas las poligonales que definen los perfiles transversales asociados a cada intervalo, éstas servirán de base para la generación de los módulos que podrán ser reutilizados en diversos puntos del escenario. Cada módulo implicará como siempre un perfil transversal de inicio y de fin. La curva de transición que se elija entre ambos perfiles será una característica de cada familia. Generalmente será una curva polinómica de tercer grado.

[296] Anexo 3.

Cada módulo codifica en su nomenclatura mediante dos dígitos, los dos perfiles transversales que definen sus extremos. Dos módulos serán compatibles longitudinalmente si comparten el mismo perfil extremo. Cada perfil implicará un determinado incremento de cota z en sentido transversal. Especial atención requerirá el nivel de proximidad más próximo a la trayectoria circulatoria (vía o carretera) por su implicación funcional. Los entornos pertenecientes a este nivel quedarán estructurados en una secuencia de desmontes o trincheras, terraplenes y llanos (en línea) en función del desnivel de alturas de sus perfiles transversales (Figura 4.65).

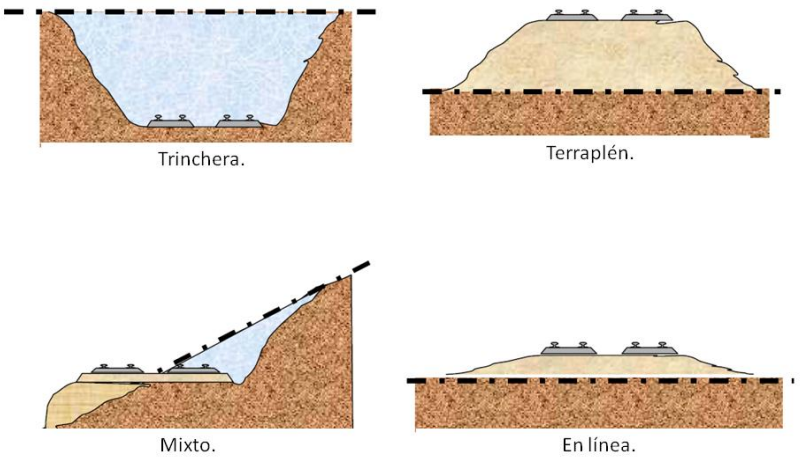


Figura 4.65. Ejemplos de posibles configuraciones del terreno.

Los incrementos de cota z de cada perfil deberán ser almacenados en una tabla, de manera que el nivel de proximidad 1 se posicionará a la misma cota z que la línea directriz y el nivel de proximidad 2, podrá posicionarse teniendo en cuenta la cota z de la línea directriz y el incremento de cota z del perfil.

4.6.2 SÍNTESIS DE PERFILES TRANSVERSALES EN LA CONSTRUCCIÓN DE ENTORNOS FICTICIOS.

La definición de la secuencia de módulos que definen un terreno se realiza a través de un conjunto de algoritmos proporcionados por la Tecnología Modular, cuya misión es garantizar el máximo aprovechamiento de los módulos, combinándolos de todas las formas posibles para aportar aleatoriedad y realismo al entorno. Al conjunto de módulos que definen un terreno se le llama *paisaje*. Son varios los parámetros a fijar para definir un tipo de paisaje:

- **Desnivel modular**: hace referencia a la variación de altura del módulo en sentido longitudinal. En función de la brusquedad del desnivel, se han establecido distintos tipos: llanura, colina, montaña, etc.
- **Niveles de proximidad abarcados**: como se comentó anteriormente el entorno se encuentra dividido en diferentes niveles de proximidad según su distancia transversal a la vía. Es necesario definir cuántos niveles se desean abarcar.
- **Aspecto**: hace referencia a las características superficiales que vendrán definidas a través de la textura. Se elegirá uno de los posibles aspectos que ofrece la base de datos: arenoso,

césped verdoso, césped seco, arcilloso, etc. Cada aspecto podrá estar formado por una o varias texturas.

- **Tipo de generación**: La Tecnología Modular ha clasificado estos paisajes en dos grupos:
  - Paisajes simétricos: los módulos se definen simétricamente respecto del punto central del intervalo.
  - Paisajes no simétricos:
    - Aleatorios: combinan todos los módulos de una familia aleatoriamente, preocupándose únicamente de garantizar la compatibilidad con las familias adyacentes.
    - Crecientes/decrecientes: combinan los módulos de manera que la cota nunca deje de crecer/decrecer. También existe la posibilidad de introducir como parámetros la cota z que se quiere alcanzar para un determinado punto kilométrico.
- **Secuencia explícita de perfiles**: en lugar de emplear los algoritmos de generación automática, existe la posibilidad de definir explícitamente la secuencia de perfiles que definen un paisaje. Cada modulo codifica sus perfiles extremos a través de dos dígitos reservados en su nomenclatura. La definición de esta secuencia, implicará la enumeración de dichos dígitos. El editor de perfiles permite visualizar los perfiles transversales existentes.

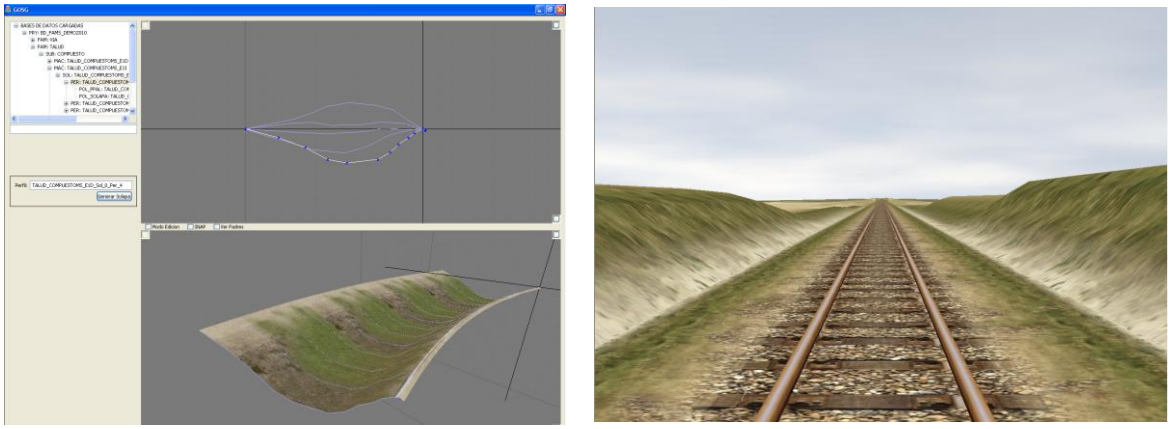


Figura 4.66. Ejemplo de una familia de terreno en la creación de un entorno ficticio.

5.LA TECNOLOGÍA MODULAR III: MODELADO FÍSICO.

5.1 INTRODUCCIÓN.

En este Capítulo la presente Tesis pretende dar solución a la *tercera fase de construcción* de un entorno virtual, afrontando la problemática asociada al posicionamiento, jerarquización y optimización del entorno virtual.

Los diferentes algoritmos de adecuación del nivel de detalle empleados en la representación en tiempo real de grandes entornos virtuales realizan discretizaciones del escenario, empleando para ello distintas primitivas de trabajo. En función de la primitiva elegida, se define una algorítmica de posicionamiento y jerarquización escénica cuyo objetivo será conseguir el acoplamiento de todas las primitivas garantizando un aspecto final continuo tanto temporalmente, es decir, sin transiciones bruscas de nivel de detalle (*popping*), como espacialmente.

En esta Tesis, los módulos, primitiva de trabajo empleada por la Tecnología Modular, serán posicionados a lo largo de una serie de líneas directrices del entorno calculadas a partir de las trayectorias circulatorias. La infinitud de posibles trazados geométricos a representar por dichas líneas directrices entra en conflicto con la finitud de módulos disponibles. Es necesario por tanto, una Algorítmica de Posicionamiento que mediante el ensamblado y repetición de un número finito de patrones pueda reproducir el entorno garantizando siempre un aspecto visual correcto sin discontinuidades espaciales.

La continuidad temporal quedará garantizada mediante una adecuada jerarquización en el grafo de la escena (*scene graph*). La algorítmica desarrollada para llevar a cabo el posicionamiento y jerarquización se ejecuta en la fase de precarga de la representación virtual, de manera que no sólo no repercuta en la velocidad de renderizado sino que facilite la creación diversos tipos de escenarios durante una sesión de entrenamiento en el simulador de conducción.

5.2 ALGORITMOS DE ADECUACIÓN DEL NIVEL DE DETALLE: POSICIONAMIENTO, JERARQUIZACIÓN Y OPTIMIZACIÓN ESCÉNICA.

Como se comentó en el Capítulo 3, los niveles de detalle continuos que emplean como primitiva de trabajo el triángulo, requieren un gran consumo de recursos e infrautilizan las capacidades de las actuales GPUs. Esto los hace desaconsejables para entornos destinados a simulaciones de conducción terrestre. En su lugar, los últimos avances en GPUS han propiciado la utilización del bloque de triángulos (*batch o cluster*) como nueva primitiva de trabajo que, generado en tiempo de precarga, garantiza la disminución de la carga de trabajo de la CPU.

En este tipo de algoritmos el terreno se divide en una estructura espacial (retícula [297], quadtree, bintree, octree), en la que el nodo perteneciente al nivel de detalle cero, la raíz, representa la versión más simplificada del terreno y el nivel de detalle último, representa la versión más detallada del mismo. Cada nodo hijo almacena una submalla del nodo padre, más detallada que la de éste y más reducida en dimensiones. La unión de todos los nodos en cada nivel de detalle define una versión completa del terreno. Para cada bloque se calcula en tiempo de precarga además de su geometría su error geométrico y su volumen de abarque, lo que permitirá en tiempo de ejecución definir su error en pantalla y determinar así si su nivel de detalle es suficiente. Si lo es, se renderiza, si no lo es se desciende recursivamente por el árbol hasta encontrar el nivel de detalle deseado.

La siguiente figura muestra un ejemplo del grafo de la escena resultado de emplear como estructura espacial un quadtree. Cada nodo del grafo representa un bloque de triángulos generado en tiempo de precarga.

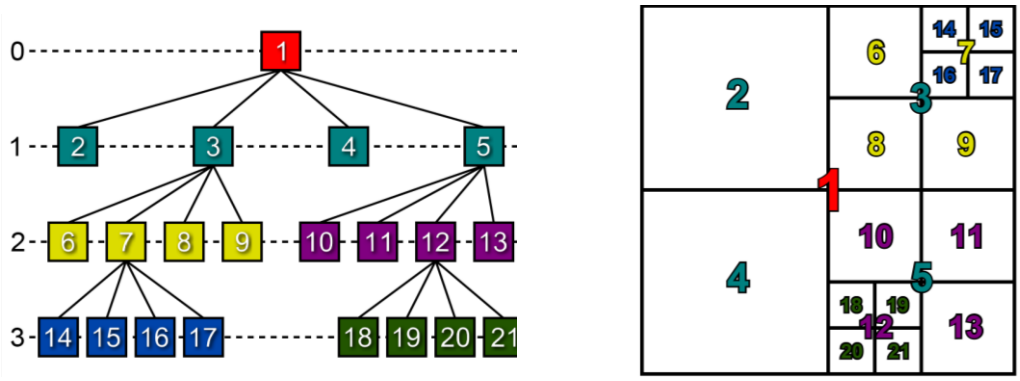


Figura 5.1.Modelo del terreno empleando un quadtree y bloques de triángulos.

Las discontinuidades espaciales que pueden aparecer en la unión de estos bloques de triángulos pueden ser agujeros y t-vértices comúnmente conocidos como cracks y t-junctions respectivamente (Figura 5.2). Estos aparecen debido a que zonas adyacentes pueden diferir en más de un grado de resolución.

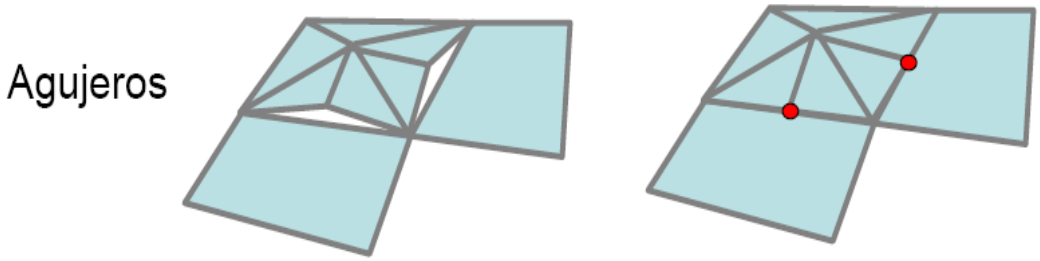


Figura 5.2.Agujeros (cracks) y T-Vértices (T-junctions).

Las formas en que estos algoritmos garantizan la continuidad espacial son muy variadas. Uno de los primeros algoritmos en emplear esta tecnología de bloques es RUSTIC [298]. Éste

[297] En realidad, el empleo de una estructura espacial reticular en la que cada celda de la retícula viene definida por un bloque de triángulos calculados en tiempo de precarga es la degeneración del nivel de detalle continuo (*continuos LOD*) en un nivel de detalle discreto (*discrete LOD*). Realmente este tipo de algoritmos que emplean bloques de triángulos precalculados son híbridos entre el modelo de LOD continuo y discreto, siendo la estructura reticular el caso límite que lo encasilla en este último grupo.

[298] Pomeranz, A. Roam using surface triangle clusters (RUSTIC). Master's thesis, University of California at Davis, 2000.

modifica el algoritmo de ROAM de manera que el triángulo que en dicho algoritmo actúa como nodo hoja del árbol binario (*bintree*), en el algoritmo de RUSTIC se convierte en un bloque de triángulos calculados en tiempo de precarga (Figura 5.3)

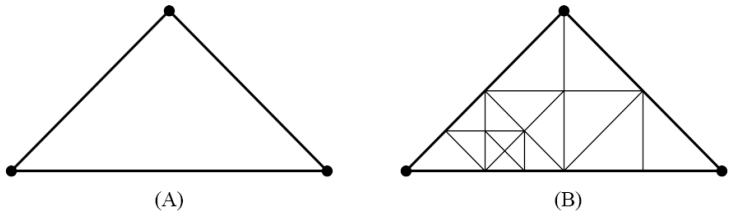


Figura 5.3.A) Triángulo en ROAM B) Triángulo en RUSTIC: el triángulo de ROAM se convierte en un bloque de 16 triángulos calculados en tiempo de precarga.

Para garantizar que la unión entre bloques se realiza sin agujeros éstos deben poseer idénticos bordes. La siguiente figura muestra dos ejemplos de construcción correcta e incorrecta de bloques.

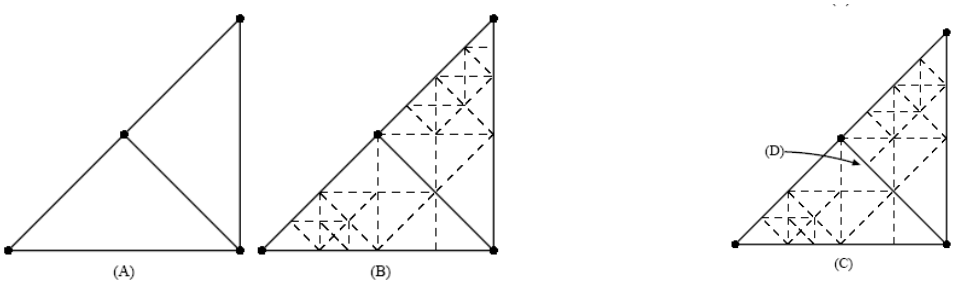


Figura 5.4.A) Dos triángulos adyacentes en ROAM B) Triángulos asociados en RUSTIC con triangulación correcta C) Triángulos asociados en RUSTIC con triangulación incorrecta ocasionando un agujero en la malla final.

RUSTIC propone diversos métodos para crear estos bloques de triángulos con esta restricción de bordes empleando divisiones (*splits*) recursivas de triángulos (Figura 5.6) y operaciones de fusión (*merge*) de estructuras diamante.

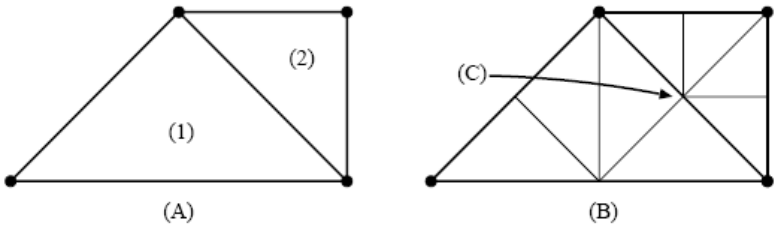


Figura 5.5.Operación de división (split) en la generación de bloques de triángulos RUSTIC: los bloques 1 y 2 comparten el split C en su borde evitando así un posible agujero en su unión.

La Figura 5.6 muestra en su centro un bloque de triángulos formando una estructura diamante, en la que (A) es una estructura diamante interna al bloque, que al no interferir con sus bordes puede ser sometida a una operación de fusión (*merge*) Por el contrario, la estructura

diamante (B) al cortar los bordes del bloque, no puede ser modificada con el fin de no cambiar los bordes del mismo.

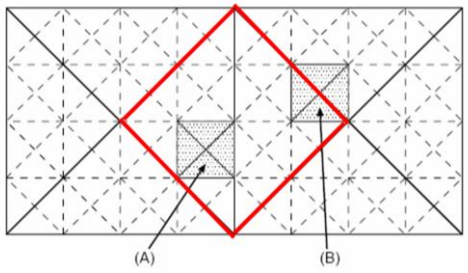


Figura 5.6.Fusión de estructuras diamante en RUSTIC.

Cignoni et al 2003 [299], en su algoritmo BDAM , posteriormente mejorado en P-BDAM [300] y C-BDAM [301], emplean también un sistema de bloques estructurado en un árbol binario (*bintree)* en el que cada bloque es una malla irregular de triángulos (TIN) calculada en tiempo de precarga. La correcta unión entre bloques viene garantizada por la definición de una función de distribución del error que asigna un error e_{k+1} en los dos bordes de menor longitud de cada bloque (error correspondiente al siguiente nivel de refinamiento del *bintree*) y un error e_k en el resto del bloque.

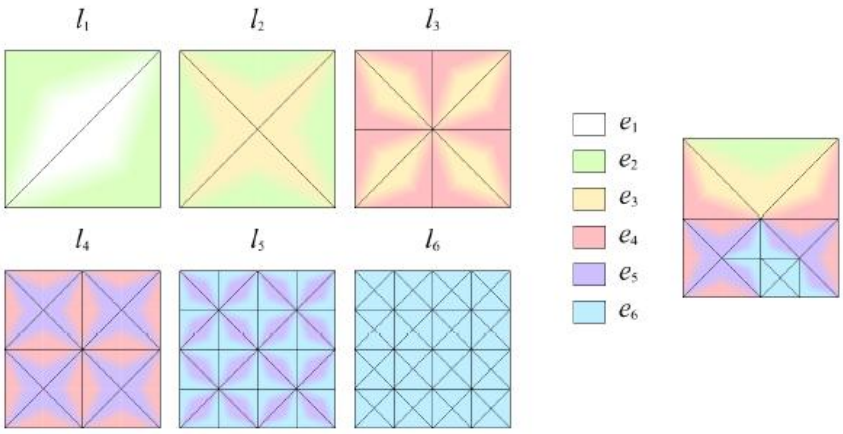


Figura 5.7.Construcción de bloques en BDAM: cada triángulo representa un bloque y está compuesto a su vez por una TIN. Cada error lleva asociado un color diferente.

Los tiempos requeridos por BDAM para calcular en tiempo de precarga los bloques de triángulos (Figura 5.8) son excesivamente elevados si lo que se pretende es su uso en una sesión de entrenamiento en un simulador de conducción en el que se busca generar y representar entornos con agilidad.

[299] Cignoni, P. et al. BDAM – batched dynamic adaptive meshes for high performance terrain visualization. Computer Graphics Forum 2003. Vol. 22, n. 3, pp. 505-514.

[300] Gobbetti, E. et al. C-BDAM - Compressed Batched Dynamic Adaptive Meshes for Terrain Rendering. In Proceedings Computer Graphics Forum, 2006. Vol. 25, n. 3, pp. 333-342.

[301] Cignoni, P., Ganovelli, F., Gobbetti, E., Marton, F., Ponchio, F., Scopigno, R.: Planet-Sized Batched Dynamic Adaptive Meshes (P-BDAM). In IEEE Visualization 2003. Pp 147-154.

Size	Tris	Time (h:m:s)	Output size	RAM
1K x 1K	2M	6:35	2 x 14MB	9MB
4K x 4K	32M	1:42:33	2 x 196MB	30MB
8K x 8K	128M	6:39:23	2 x 765MB	115MB

Figura 5.8.Requisitos de tiempo y memoria en el cálculo de bloques de triángulos en BDAM (PC con dos procesadores AMD Athlon MP 1600 MHz, 2GB de RAM, tarjeta gráfica NVIDIA GeForce 4 Ti4600)

Lario et al 2003 [302], presentan una metodología que llaman de hiperbloque, en la que el terreno es estructurado en un quadtree formado por bloques que almacenan TINs a distintas resoluciones generadas en tiempo de precarga. En tiempo de ejecución se selecciona la resolución deseada para cada bloque (Figura 5.9 a), se calculan las transiciones de bloque para que no existan discontinuidades (Figura 5.9 b) y finalmente se genera la malla continua (Figura 5.9 c).

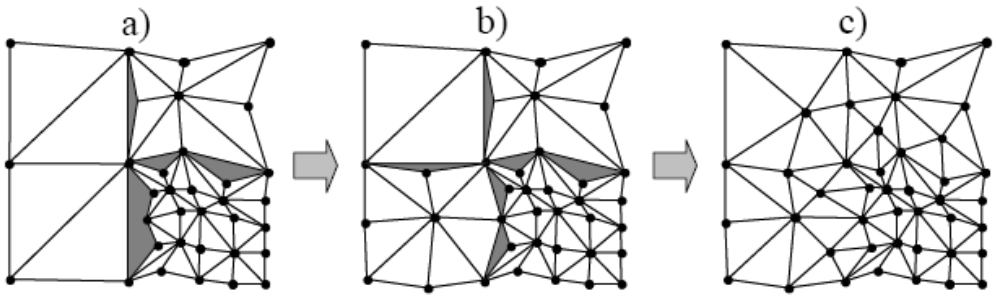
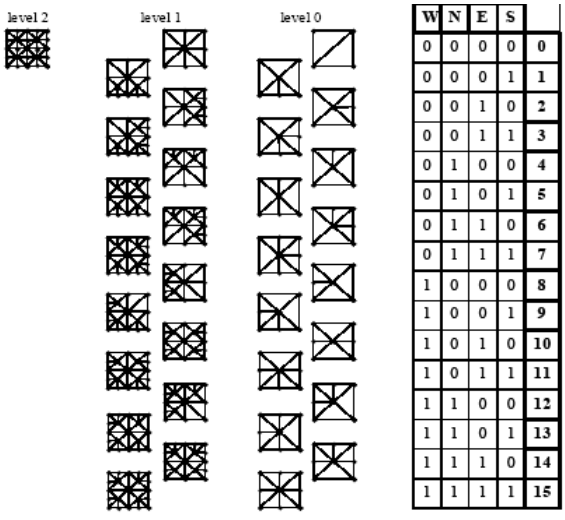


Figura 5.9.Proceso de triangulación y unión de bloques en Lario et al 2003.

Para evitar agujeros en la unión entre bloques, las transiciones entre los mismos se generan atendiendo a una tabla (Figura 5.10), en la que se muestra la triangulación resultante de la inserción de nuevos vértices en cada uno de los cuatro bordes de un hiperbloque, representados por los cuatro puntos cardinales.



W	N	E	S	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

Figura 5.10.Tabla de bordes y la correspondiente triangulación para los niveles 0, 1 y 2. W,N, E y S se corresponden con los bordes, oeste, norte, este y sur respectivamente.

[302] Lario, R., Pajarola, R., Tirado, F. Hyperblock-QuadTIN: Hyper-block quadtree based triangulated irregular networks. In IASTED International Conference on Visualization, Imaging and Image Processing, VIIP 2003.

Pouderoux et al 2005 [303] dividen el terreno en bloques a los que asocian un modelo multiresolución consistente en el empleo de diferentes *strip masks*. Un *strip mask* es una triangulación regular del bloque a una determinada resolución (Figura 5.11). Su algoritmo permite adaptar la renderización del terreno a distintos tipos de hardware, desde una PDA a un PC.

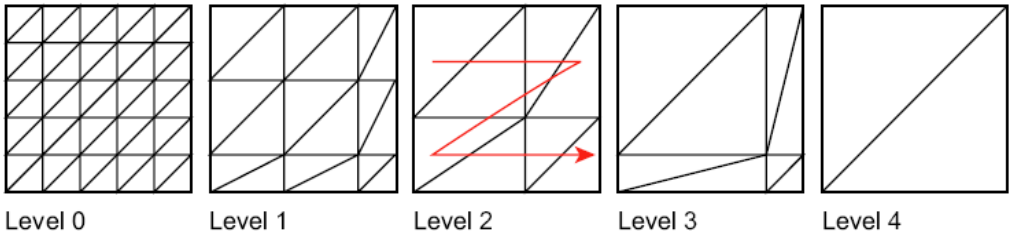


Figura 5.11.Bloque a distintas resoluciones en Pouderoux et al 2005.

Dado que cada bloque se triangula de manera independiente surgen agujeros (*cracks*) en sus fronteras. Para disimularlos el algoritmo emplea planos con texturas difusas bajo el terreno (Figura 5.12).

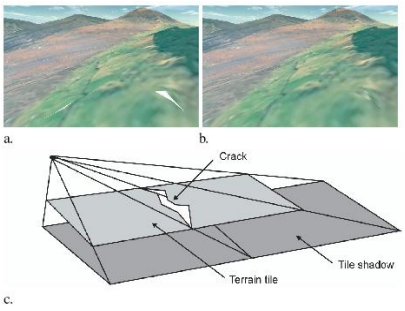


Figura 5.12.Atenuación de los huecos entre bloques de distinta resolución mediante el empleo de planos de textura.

Livny et al 2009 [304] dividen el terreno en bloques rectangulares de diferentes resoluciones (Figura 5.13)

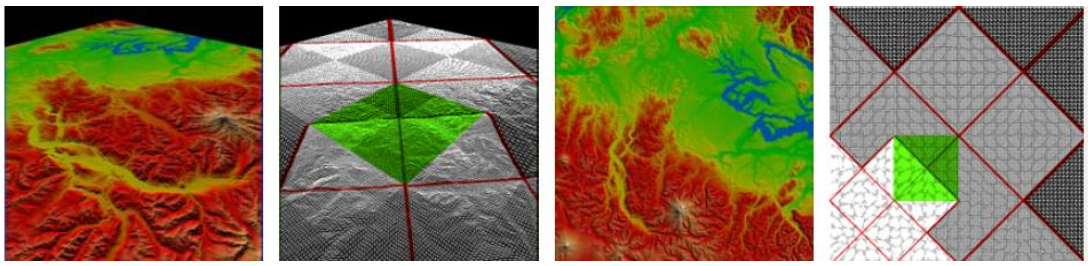


Figura 5.13.Renderizado del terreno mediante el algoritmo de Livny et al 2009: bloque coloreado en verde.

[303] Pouderoux, J. and Marvie, J. Adaptive streaming and rendering of large terrains using strip masks. In Proceedings of the 3rd international Conference on Computer Graphics and interactive Techniques in Australasia and South East Asia. GRAPHITE '05. ACM Press, New York, NY,.Vol .8. pp 299-306.

[304] Livny, Y., Kogan, Z. and El-Sana, J. Seamless patches for GPU-based terrain rendering. In Proceedings of The Visual Computer 2009. pp 197-208.

Cada bloque está constituido por cuatro losetas triangulares, cada una de las cuales puede tener una resolución diferente y por cuatro rebordes (Figura 5.14).

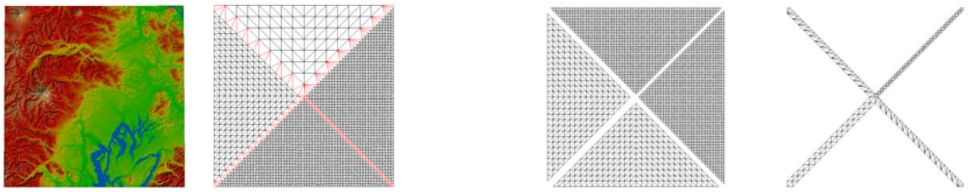


Figura 5.14.Imagen de un bloque de triáqngulos y su visualización alámbrica: losetas triangulares y rebordes.

Los rebordes sirven de transición entre las losetas, de manera que su diferente resolución no origine ningún tipo de discontinuidad (Figura 5.15).

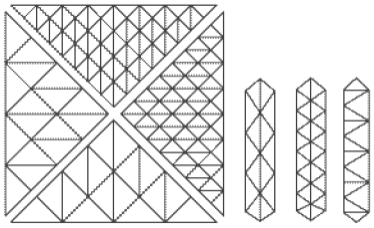


Figura 5.15.Losetas triangulares a dos resoluciones distintas y sus correspondientes rebordes.

Smartmesh, patentado por TERREX y parte integrante de Terra Vista [305] ofrece una interesante propuesta para evitar solapes incorrectos entre los bloques. Tradicionalmente en los sistemas de LODs discretos el problema de aparición de huecos en las uniones de dos bloques con diferente nivel de detalle se garantizaba forzando a que los vértices en los bordes de los mismos permanecieran intactos en todos los niveles de detalle. Esto provocaba un efecto visual incorrecto en las uniones de bloques como consecuencia de su resolución constante, independiente de la resolución interna del bloque, como se puede observar en la Figura 5.16.

Figura 5.16.Problemas visuales ocasionados en los bordes de los bloques, como consecuencia de la constancia de vértices a lo largo de éstos en todos los niveles de detalle.

[305] http://www.presagis.com/products_services/products/modeling-imulation/content_creation/terra_vista/ [Consulta: 12 Enero 2013]

La solución ofrecida por Smartmesh consiste en dividir cada bloque en cinco zonas (Figura 5.17). La zona interior se triangula en función del nivel de detalle deseado para el bloque y las zonas laterales se triangulan garantizando la continuidad con todos los bloques limítrofes.

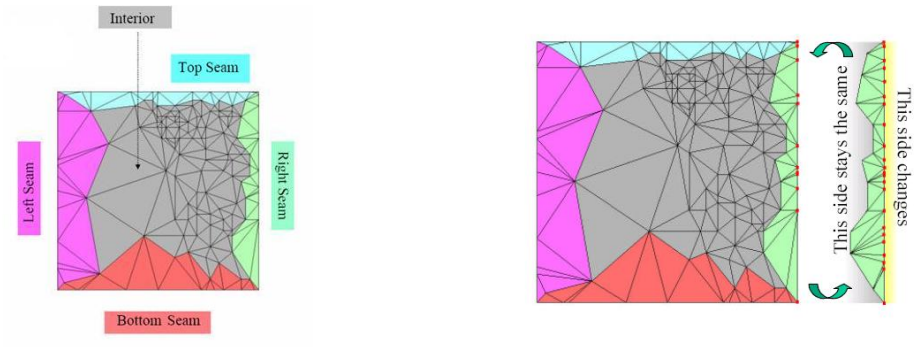


Figura 5.17.Solución ofrecida por SmartMesh.

En ninguno de los algoritmos vistos se trata la representación de elementos lineales.

La presente Tesis garantiza mediante su sistema de instanciación una representación virtual con continuidad temporal y espacial gracias a su algorítmica de Posicionamiento Modular, la deformación de módulos mediante shaders (o en su defecto mediante las solapas) y su sistema de jerarquización escénica.

La algorítmica de Posicionamiento Modular será responsable de la adecuada elección y colocación del tipo de módulo a insertar en cada punto kilométrico del entorno, garantizando un aspecto visual correcto y la flexibilidad constructiva necesaria en este tipo de entornos.

Su sistema de jerarquización escénica, mediante el empleo de un sistema de niveles de detalle discretos que busca minimizar el consumo de CPU, permite alcanzar las elevadas velocidades de refresco exigidas a los simuladores de conducción terrestre.

Finalmente la incorporación de shaders que deforman la geometría potencia las ventajas de este sistema de instanciación, minimizando la memoria de almacenamiento necesaria y los tiempos de carga y maximizando la versatilidad constructiva.

5.3 ALGORITMICA DE POSICIONAMIENTO MODULAR.

Los algoritmos de Posicionamiento Modular desarrollados por esta Tesis tienen como objetivo generar entornos realistas y variados, con continuidad espacial y tangencial, solventando para ello las restricciones impuestas por un sistema de instanciación formado por un número finito de elementos modulares.

Estos algoritmos parten de una definición geométrica base (biarco, NURBS, nube de puntos) dividida en intervalos, cada uno de los cuáles estará asociado a una familia modular. En función de la parte del escenario que se esté generando, dicha división en intervalos está motivada por diversas causas. En el caso por ejemplo de querer representar virtualmente una trayectoria circulatoria, como puede ser una vía de un entorno ferroviario o una carretera de un entorno urbano, dichas divisiones estarán generadas por la existencia de secciones circulatorias, que a su vez, como se vió en el Capítulo 4, se generan debido a la existencia de diversos tipos de nodos circulatorios. Con el fin de permitir que cada sección pueda presentar un aspecto visual diverso al de sus colindantes, cada una de ellas constituye un intervalo independiente que llevará asociada una familiar modular determinada. En el caso de representar virtualmente un entorno propiamente dicho, como puede ser un terreno, la división estará motivada por la existencia de

diversos accidentes geográficos, o en el caso de un túnel de una línea ferroviaria, por la existencia de distintos tipos de túneles. De esta manera cada intervalo vendrá caracterizado por una definición geométrica base y una familia modular.

Los algoritmos de Posicionamiento Modular desarrollados por esta Tesis han de resolver los siguientes puntos:

- **El tipo de módulo** a colocar en cada punto del escenario: si la curvatura del módulo no es la correcta se producen discontinuidades espaciales y de tangencia. El efecto visual que produce la introducción de un módulo de graduación incorrecta lo podemos ver en las siguientes figuras. La Figura 5.18 muestra el aspecto visual que resulta de introducir un módulo de raíl más curvado de lo que debiera.

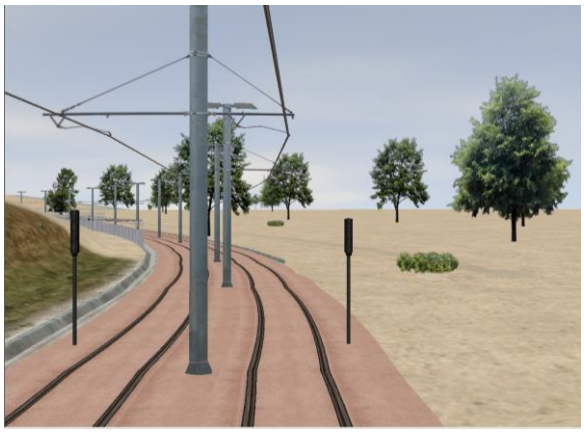


Figura 5.18.Sobrecurvado de módulos de raíl.

La Figura 5.19 muestra el aspecto visual que resulta de introducir un módulo de raíl menos curvado de lo que debiera.

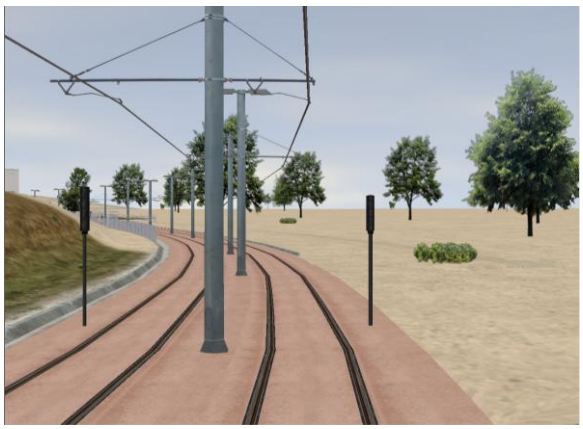


Figura 5.19.Subcurvado de módulos de raíl.

- **El escalado de cada módulo**. Los módulos podrán ser sometidos a un escalado en X, en Y y/o en Z. El escalado máximo admisible variará en función de la longitud básica del módulo, *Lb*. El valor máximo de este escalado se establece experimentalmente. En el caso de un entorno ferroviario, para una *Lb*=20 m, este valor es del 10%. Superar esta magnitud supone por un lado la deformación excesiva de la solapa y de la textura. La Figura 5.20 muestra el aspecto visual de un trazado construido con módulos sometidos a un escalado excesivo.

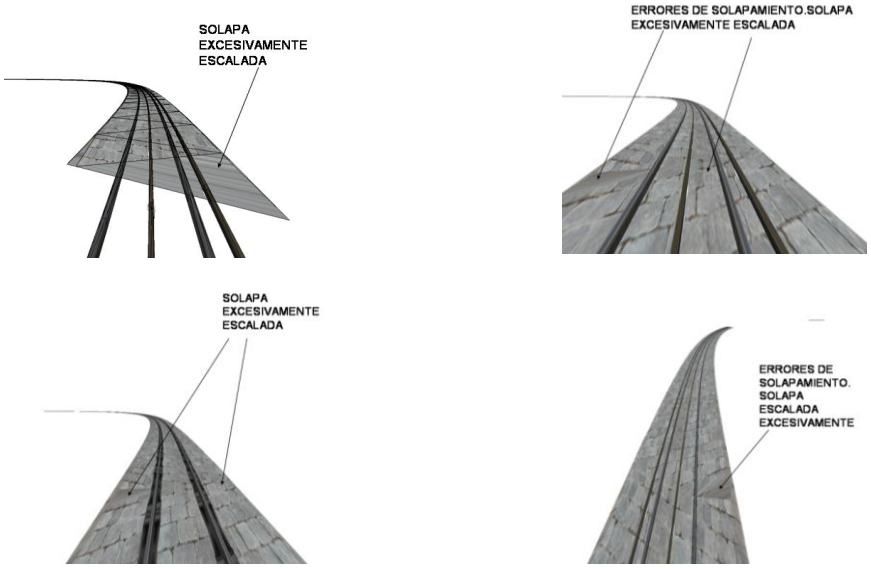


Figura 5.20.Problemas surgidos como consecuencia de un escalado excesivo de módulo.

Por otro lado, un escalado excesivo supone la variación de la curvatura del módulo de una manera notable, dando lugar a un incorrecto acoplamiento de módulos. Únicamente podrán superarse estas tolerancias de escalado si se emplean shaders que corrijan dichas distorsiones. En este caso, la Tecnología Modular calculará dichos parámetros de deformación, que el motor gráfico se encargará de manejar en tiempo de ejecución.

- **Las coordenadas de posicionamiento de cada módulo**. La posibilidad de manejar varios formatos que definan la geometría de las líneas directrices del entorno (biarco, NURBS y nube de puntos), ha motivado la creación de diferentes algoritmos de posicionamiento de módulos. A continuación se explican en detalle cada uno de ellos y las condiciones idóneas para su aplicación. La elección en cada momento del método de posicionamiento más idóneo no sólo viene condicionada por la garantía de un aspecto visual correcto sino que ha de facilitar el intercambio de información entre los diversos subsistemas involucrados en la simulación, dado que las trayectorias circulatorias no sólo actúan como líneas directrices para el posicionamiento modular sino que también han de guiar el tráfico.

5.3.1 ALGORITMO CIRCUNSCRITO.

El Algoritmo Circunscrito se aplica en aquellas ocasiones en las que la definición geométrica de base es un biarco. El lugar geométrico de posicionamiento de módulos se denomina *poligonal* y consiste en una serie de puntos obtenidos mediante interpolación sobre dicho biarco.

Los módulos se posicionan en el punto medio de su extremo izquierdo y orientan su cuerda según el segmento de la poligonal sobre el que asientan, abarcando el ángulo de arco comprendido entre ambos extremos de su cuerda. Así los módulos circunscriben la línea directriz, según se muestra en la siguiente figura.

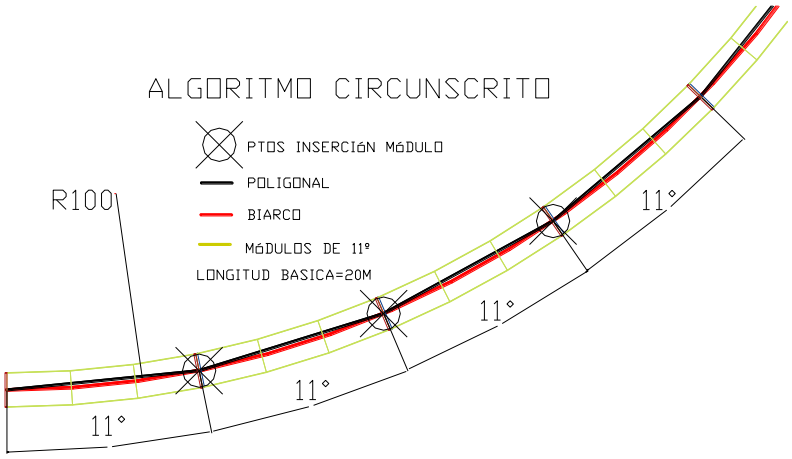


Figura 5.21.Algoritmo Circunscrito: poligonal y módulos.

La Figura 5.22 muestra la colocación de una serie de módulos de plataforma ferroviaria y raíl según el Algoritmo Circunscrito. En la izquierda aparece cada módulo con un color distinto para poder observar con más claridad el correcto acoplamiento entre ellos, ya que en la imagen de la derecha la perfección del acoplamiento hace imposible discernir dicha unión.

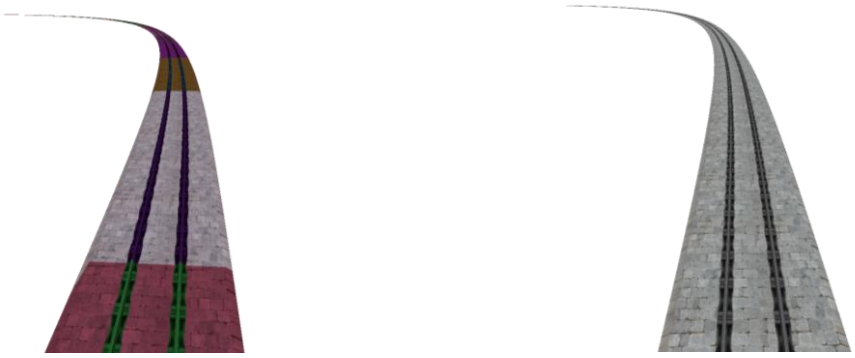


Figura 5.22.Módulos de plataforma ferroviaria y módulos de raíl colocados según el Algoritmo Circunscrito.

La distancia entre puntos de la poligonal se determinará teniendo en cuenta la longitud del módulo, llamada *longitud básica* (*Lb*) y las tolerancias de deformación aplicadas. Existen una serie de puntos de obligado paso que definirán la longitud de línea directriz, *L'*, que habrá de cubrirse con un determinado número entero de módulos, *N*. Estos puntos de obligado paso son:

a. los inicios y fines de arco, puesto que los módulos poseen un radio constante y no pueden abarcar más de un arco;

b. los inicios y fines de intervalo, por definir la transición entre dos familias modulares distintas.

Puesto que dichas longitudes *L'* no tienen porque ser múltiplos de la longitud básica, será necesaria la inserción de módulos especiales que salven estas discrepancias. Dichos módulos especiales podrán ser módulos generados ad hoc o módulos rectos de longitud *L'b*, que se posicionarán siguiendo el trazado. De esta manera, sea una longitud de línea directriz *L'*, una longitud básica *Lb* y un máximo escalado permitido *e* tales que:

*L'=n*Lb+μ*Lb* siendo *n*Lb<L'<(n+1)*Lb*, *n* el número de módulos de longitud básica y $0<\mu<1$.

Se trata de buscar el número *N* de módulos que define la frontera entre el uso o no uso de módulos especiales. Sea *e*, el escalado máximo admisible, entonces:

$$\epsilon_n = \frac{L'}{n*Lb} = \frac{n*Lb+\mu*Lb}{n*Lb} = 1+\frac{\mu}{n} \le 1+e \quad (1)$$

$$\epsilon_{n+1} = \frac{L'}{(n+1)*Lb} = \frac{n*Lb+\mu*Lb}{(n+1)*Lb} = 1-\frac{1-\mu}{n+1} \ge 1-e \rightarrow \frac{1-\mu}{n+1} \le e \rightarrow n+1 \ge \frac{1-\mu}{e} \quad (2)$$

Teniendo en cuenta (1) y (2), se pueden definir las longitudes de los intervalos que emplearán únicamente módulos de longitud *Lb*, o bien únicamente módulos especiales o bien mezcla de ambos tipos de módulos:

$$\left.\begin{array}{l}\mu \le e*n \\ \mu \ge 1-e*(n+1)\end{array}\right\} \longrightarrow e*n = 1-e*(n+1) \longrightarrow n = \text{int}\left(\frac{1-e}{2*e}\right)+1$$

Es decir, si la longitud de línea directriz *L'* cumple:

- *L'≥ n*Lb*(1-e)* ó *Lb*(1-e) < L'< Lb*(1+e)*, el intervalo se construirá únicamente con *N* módulos de longitud *Lb*, escalados una magnitud *e*, siendo $N = round\left(\frac{L'}{Lb}\right)$ *y* $e = \frac{L'}{N*Lb}$.
- *Lb*(1+e) < L'< n*Lb*(1-e)*, el intervalo se construirá con *N* módulos de longitud *Lb* y *N'* módulos especiales, siendo $N = round\left(\frac{L'}{Lb}\right)$, si el módulo especial es un módulo ad hoc N'=1 y si los módulos especiales son módulos rectos de longitud *Lb'*, $N' = round\left(\frac{L'-N*Lb}{Lb'}\right)$, $e' = \frac{L'-N*Lb}{N'*Lb'}$.
- *L'< Lb*(1-e)*, el intervalo se construirá con módulos especiales, si el módulo especial es un módulo ad hoc N'=1 y si los módulos especiales son módulos rectos de longitud *Lb'*, $N' = round\left(\frac{L'}{Lb'}\right)$, $e' = \frac{L'}{N'*Lb'}$.

La determinación del módulo que es necesario introducir en cada punto de la poligonal lo determinará el ángulo de abarque del biarco entre los dos puntos extremos del módulo:

$$\alpha = \frac{L_{arco}}{R}$$

siendo L_{arco} la longitud de la línea media del módulo y R el radio del biarco. El módulo elegido será por tanto aquél perteneciente al conjunto base de módulos [306] cuyo ángulo de abarque más se aproxime a α.

[306] Ver Capítulo 3 apartado 3.5.

5.3.1.1 VARIANTE DEL ALGORITMO CIRCUNSCRITO.

Cuando la definición geométrica de la línea directriz no viene expresada mediante un biarco, por ejemplo, si la poligonal se ha visto sometida a un proceso de edición o consolidación y posteriormente no se ha suavizado, el ángulo de abarque del módulo propuesto por el Algoritmo Circunscrito deja de tener sentido, al desaparecer el concepto de arco. Esto motivó el desarrollo de nuevas variantes de dicho algoritmo.

Para definir el tipo de módulo a colocar en cada segmento de una poligonal definida por n+ 1 puntos, se plantea el siguiente sistema de n+1 ecuaciones y n incógnitas:

$$\delta_0 = \frac{\theta_0}{2} - \frac{\alpha_0}{2}$$

.

.

$$\delta_j = \theta_j - \frac{\alpha_{j-1} + \alpha_j}{2}$$

$$\delta_{j+1} = \theta_{j+1} - \frac{\alpha_j + \alpha_{j+1}}{2}$$

.

.

$$\delta_n = \frac{\theta_n}{2} - \frac{\alpha_{n-1}}{2}$$

donde δ_i representa el error de solapamiento entre módulos, θ_i el incremento angular a cubrir entre módulos y α_i el ángulo del módulo a insertar. Es por tanto un sistema donde se buscan los valores de α_i que minimizan los valores de δ_i , siendo P_0 y P_n los puntos extremos de la línea directriz (Figura 5.23).

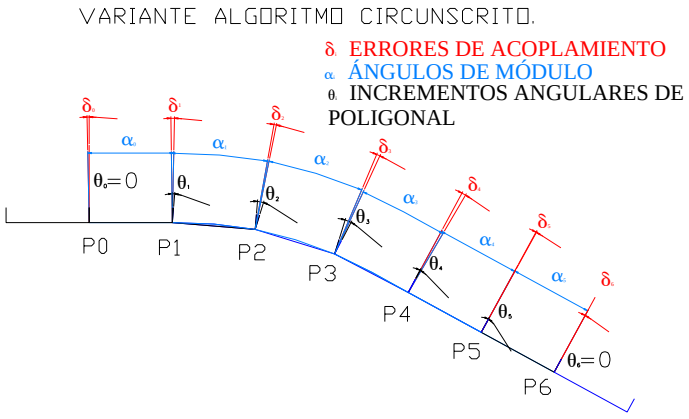


Figura 5.23.Variante del Algoritmo Circunscrito :errores al colocar módulos sobre la poligonal.

El primer intento por solventar este sistema consistió en definir el ángulo de abarque de cada módulo como la semisuma de los ángulos formados en el plano XY con sus segmentos adyacentes:

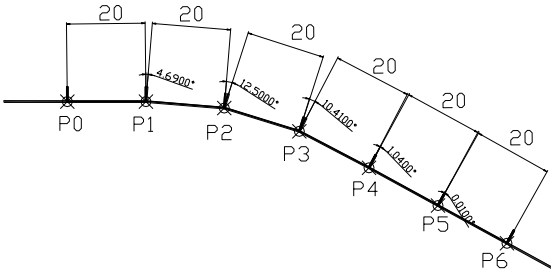
$$\alpha_{j+1} = \frac{\theta_j + \theta_{j+1}}{2}$$

De esta manera cada módulo absorbe la mitad de los desfases angulares de sus extremos y son los módulos adyacentes los que se encargan de cubrir el desfase restante.

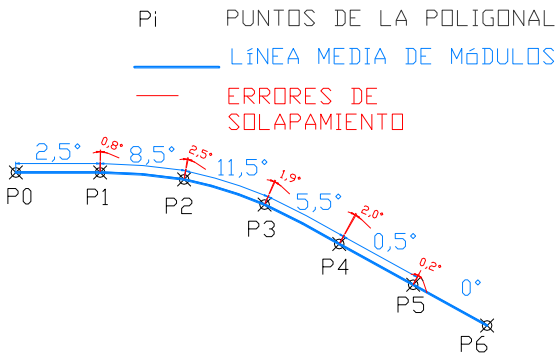
Sin embargo, la posibilidad de que entre vértices consecutivos de la poligonal existan existan variaciones angulares muy diversas da lugar a que en la determinación del módulo que es necesario insertar en un determinado punto intervengan dos semiángulos muy distintos y que como consecuencia, aparezcan solapamientos y aperturas inadmisibles. Se optó así por intentar minimizar estos errores buscando para ello una distribución más óptima de los ángulos de abarque de los módulos, lo que constituye un problema de optimización múltiple.

El gran hándicap que plantea la resolución de este problema de optimización múltiple está en que la resolución automática del mismo no garantiza siempre un aspecto visual satisfactorio. Esta limitación motiva el diseño de nuevos algoritmos de posicionamiento modular, que busquen nuevos lugares geométricos de puntos inserción de módulo.

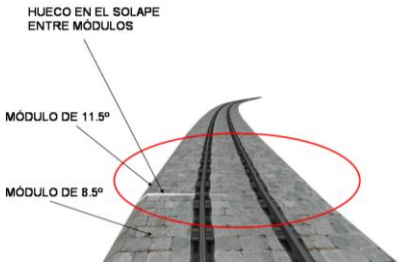
La Figura 5.24 (a) muestra un ejemplo de línea directriz editada manualmente. Como consecuencia de dicha edición la definición geométrica base deja de ser un biarco y pasa a ser una nube de puntos, que constituye la poligonal. La Figura 5.24 (b) muestra la solución ofrecida por la variante del Algoritmo Circunscrito para esta definición geométrica. La Figura 5.24 (c) y (d) muestran el aspecto visual final obtenido empleando módulos con y sin solapas.



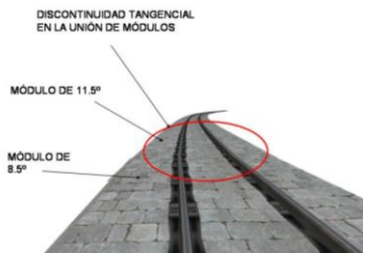
(a) Punto de la Poligonal sobre línea directriz editada manualmente.



(b) Línea media de módulos y errores de solapamiento y abertura entre los mismos.



(c) Aspecto visual final empleando módulos sin solapas: discontinuidad espacial.



(d) Aspecto visual final empleando módulos con solapas: discontinuidad tangencial.

Figura 5.24.Variante del Algoritmo Circunscrito sobre línea directriz editada .

5.3.2 ALGORITMO INSCRITO.

El Algoritmo Inscrito surge ante la necesidad de representar entornos virtuales cuya línea directriz pueda venir definida tanto con biarcos, como NURBS o nubes de puntos, algo inadmisible con el Algoritmo Circunscrito.

En este algoritmo los módulos se posicionan sobre los segmentos de la poligonal descrita en el Algoritmo Circunscrito, buscando una distribución simétrica entorno a los vértices de la misma. De esta manera los módulos se inscriben en la línea directriz y absorben los incrementos angulares que se producen en los vértices de la poligonal. El lugar geométrico de inserción de módulos se denomina en este caso *segmentación*. En el caso de que los segmentos de poligonal tengan todos igual longitud, dicho lugar geométrico coincide con los puntos medios de los segmentos de la poligonal.

Tomando el ejemplo de la línea directriz editada de la Figura 5.24, la Figura 5.25 muestra el lugar geométrico de puntos de inserción de módulo según el Algoritmo Inscrito, así como el incremento angular que ha de cubrir cada módulo.

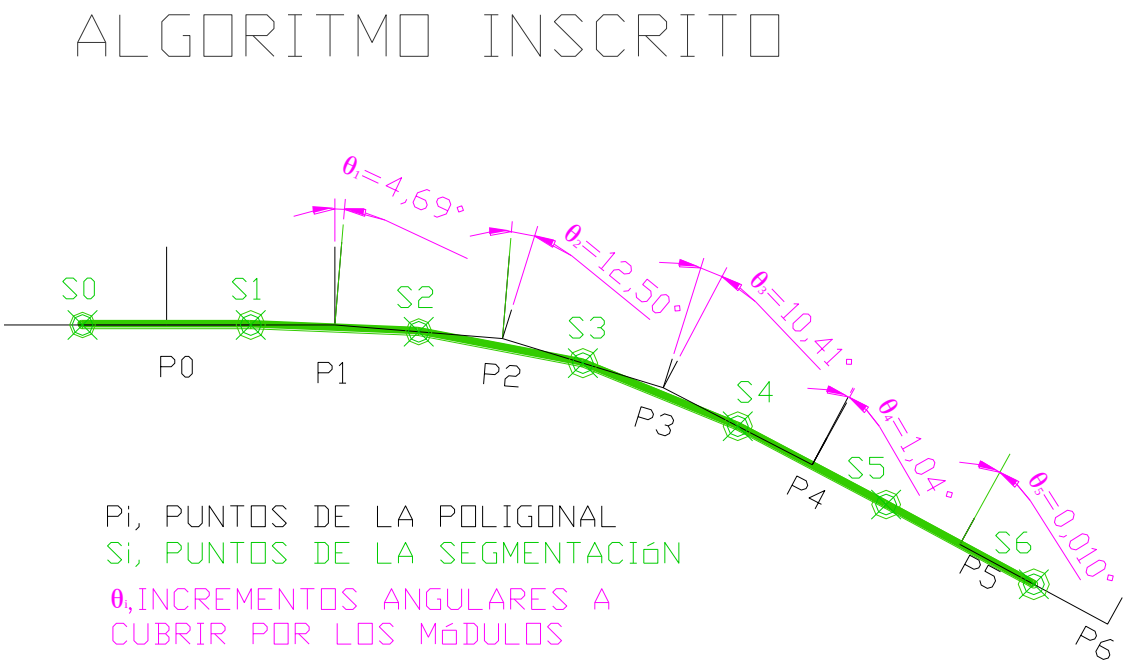
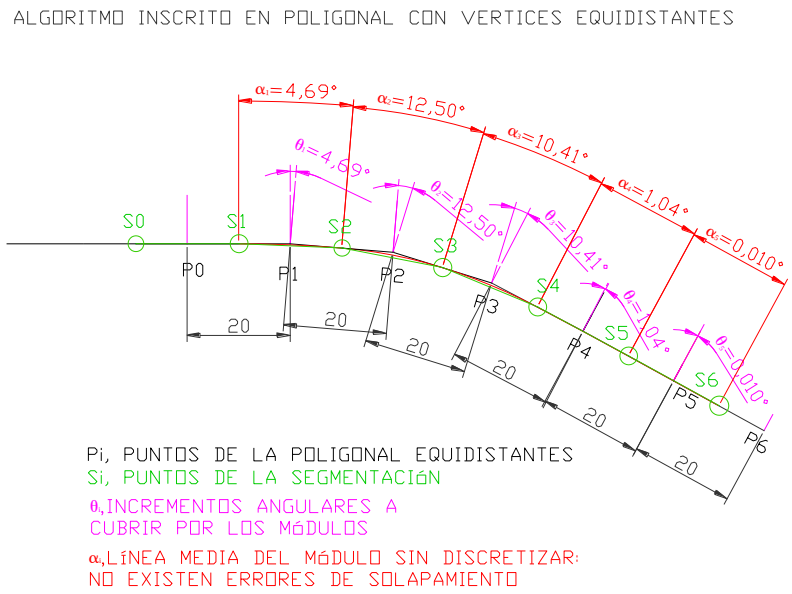
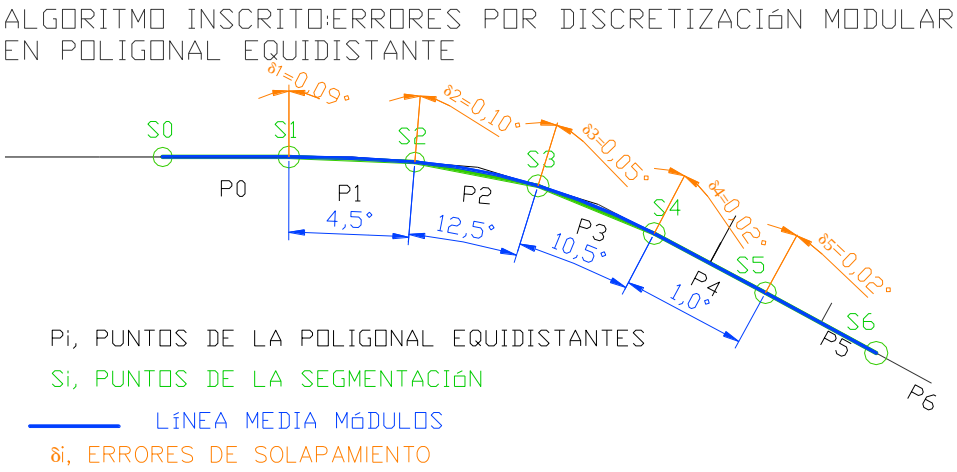


Figura 5.25.Algoritmo Inscrito: puntos de segmentación e incrementos angulares a cubrir.

El algoritmo inscrito no produce errores de solapamiento cuando opera sobre segmentos de igual longitud, ya que esto garantiza la simetría de los puntos de segmentación entorno a los vértices de la poligonal y por tanto una absorción óptima de los incrementos angulares de la poligonal (Figura 5.26 a), proporcionando la continuidad espacial y tangencial requerida. En este caso, el único error cometido por el algoritmo es el que resulta de la discretización angular de los módulos (Figura 5.26 b), error imperceptible en una representación virtual (Figura 5.26 c).



(a) Puntos de segmentación e incrementos angulares a cubrir.

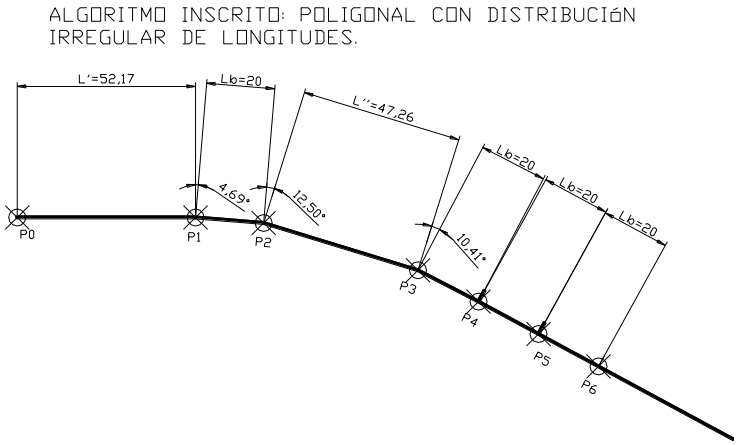


(b) Línea media de los módulos y errores de solapamiento debidos a la discretización modular.

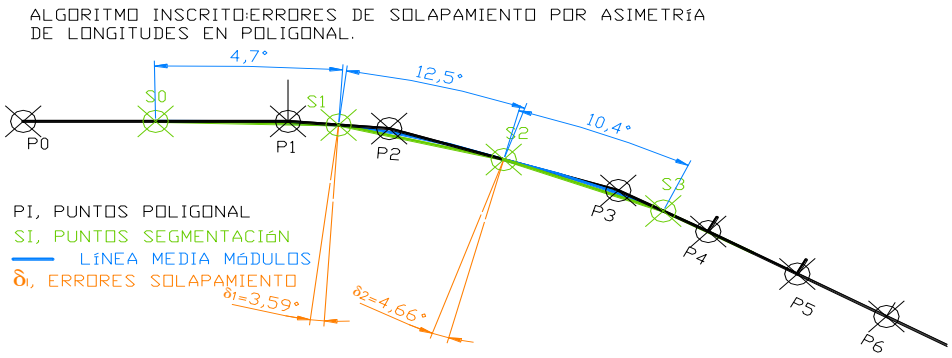
(c) Aspecto visual final.

Figura 5.26.Algoritmo Inscrito sobre línea directriz editada.

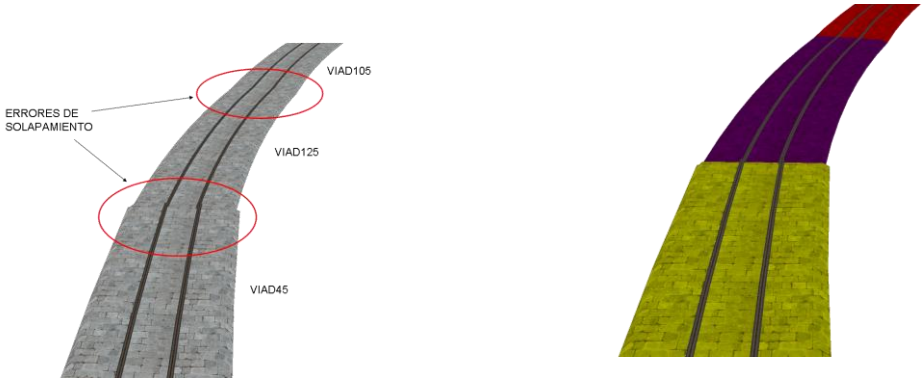
Sin embargo, cuando los segmentos de la poligonal poseen distintas longitudes (Figura 5.27 a) es necesaria la adición o eliminación de puntos de inserción que garanticen el posicionamiento simétrico de los módulos entorno a los vértices de la poligonal, evitando así errores de apertura o solapamiento (Figura 5.27 b, c).



(a) Poligonal con distintas longitudes.



(b) Errores por no tener en cuenta que la poligonal posee segmentos de distintas longitudes.



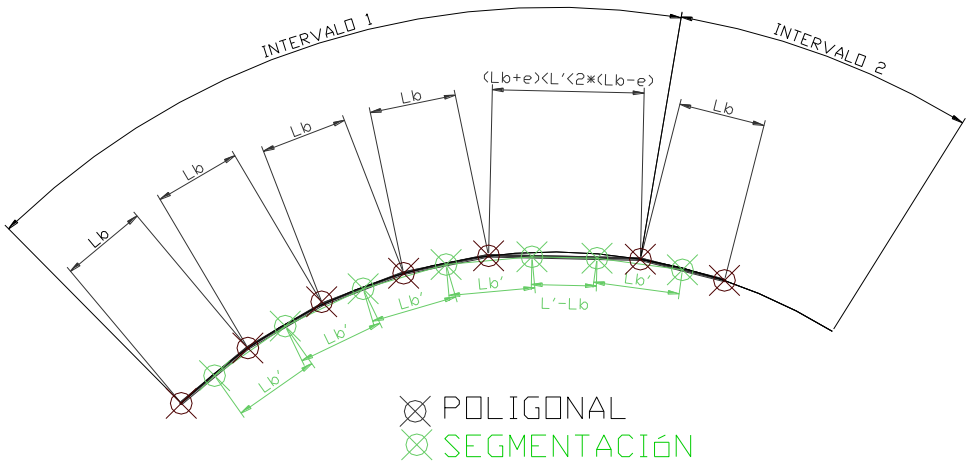
(c) Aspecto visual final

Figura 5.27.Algoritmo Inscrito en poligonal con segmentos de distintas longitudes.

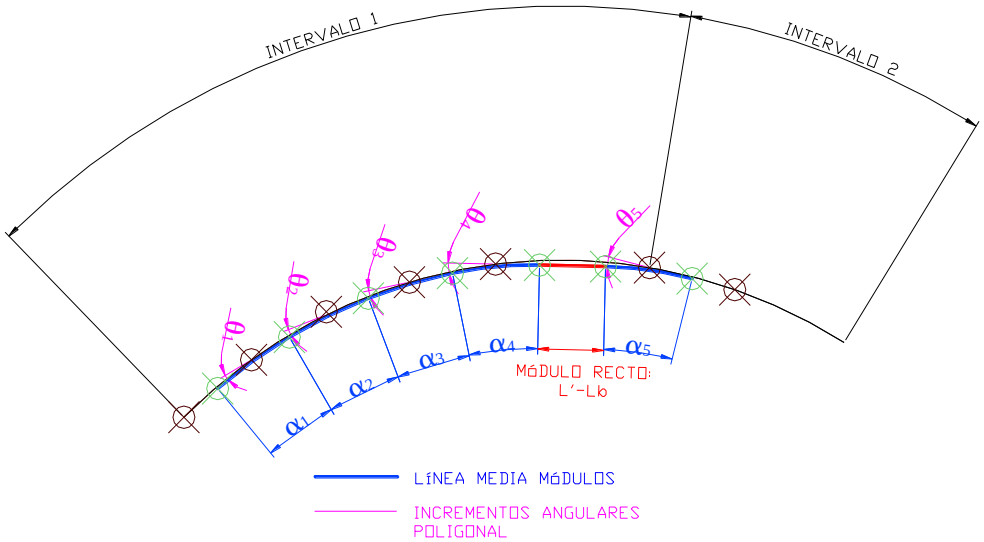
El procedimiento a seguir para calcular los puntos de inserción de módulo (*segmentación*) teniendo en cuenta la existencia de segmentos de distinta longitud es el siguiente. Sean L_i la longitud del segmento de poligonal P_iP_{i+1}, L_b la longitud básica del módulo y e el escalado máximo admisible, entonces:

- Si L_b-$e \leq L_i \leq L_b$+e, entonces el punto de inserción S_i es el punto medio del segmento de longitud L_i, es decir, P_iS_i=(P_iP_{i+1})*0.5. Este módulo absorberá el incremento angular que se produce en el vértice P_{i+1} de la poligonal.

- Si $L_b + e < L_i < (L_b - e)*2$, entonces el segmento L_i dará lugar a dos puntos de inserción de módulo (Figura 5.28). El primero, S_i, distará del extremo inicial del segmento una distancia d= L_b *0,5 y el segundo, S_{i+1}, distará del extremo final del segmento la misma distancia. El punto S_i, será el punto de inserción de un módulo especial que cubra la longitud (L_i - L_b). El punto S_{i+1} será el punto de inserción de un módulo que absorberá el incremento angular que se produce en el vértice P_{i+1} de la poligonal.
- Si $L_i < L_b$ -e, se procederá a la eliminación de un punto de la poligonal.
- Si $L_i \geq (L_b - e)*2$ se dividirá el segmento en dos partes iguales.



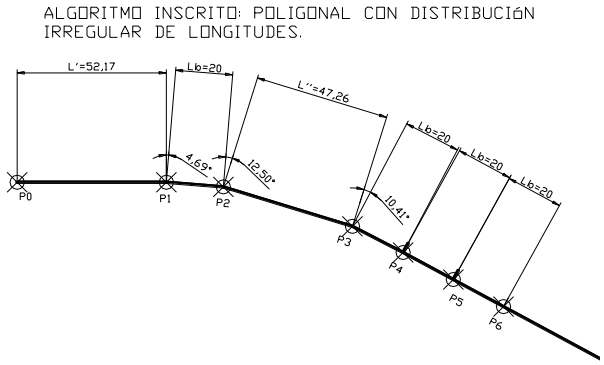
(a) Poligonal y segmentación.



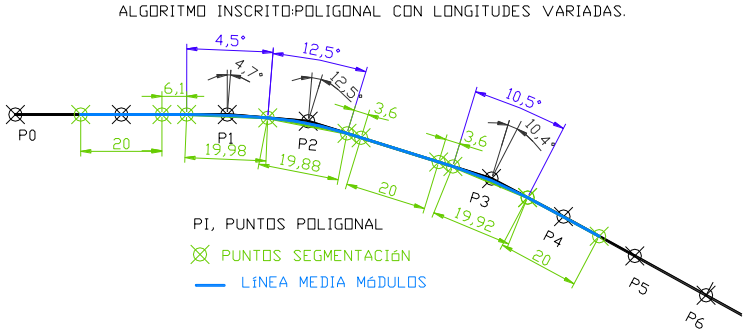
(b) Línea media de los módulos.

Figura 5.28.Algoritmo Inscrito con segmentos de la poligonal de distinta longitud.

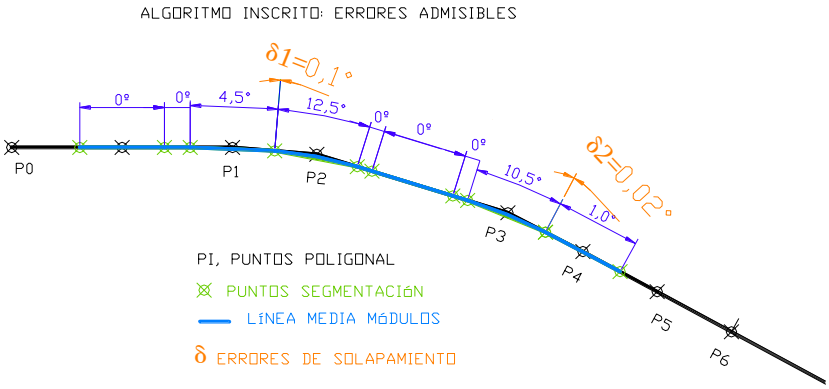
A continuación se muestra como ejemplo la solución planteada por el Algoritmo Inscrito para el caso de la línea directriz con segmentos de distintas longitud vista en a Figura 5.27(a).



(a) Poligonal con distintas longitudes.



(b) Línea media de módulos



(c) Errores resultado de la discretización modular, inadvertibles en la representación virtual.

(d) Aspecto visual final

Figura 5.29.Algoritmo inscrito en poligonal con segmentos de distintas longitudes.

Como conclusión se puede decir, que el Algoritmo Inscrito garantiza un aspecto visual correcto, sin solapamientos ni aberturas entre módulos, sin embargo obliga a que los inicios y fines de módulo vayan desfasados respecto de los inicios y fines de intervalos (secciones, entornos, etc). Esto lo hace inadecuado en situaciones donde los límites de intervalo exigen un cambio de familia modular (Figura 5.30).

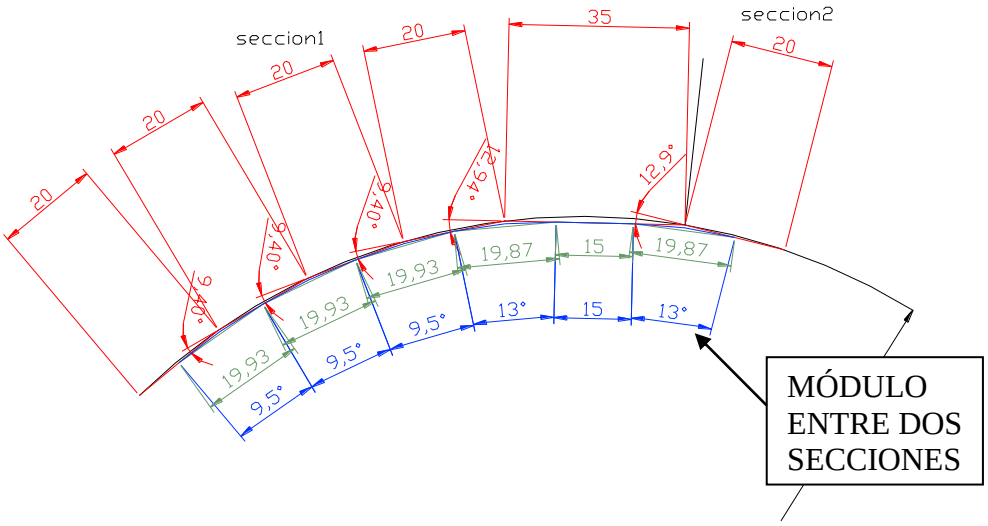


Figura 5.30.Algoritmo Inscrito: módulo abarcando dos secciones.

Teniendo en cuenta que el Algoritmo Inscrito nos permite más versatilidad en cuanto a la definición geométrica base y que el Algoritmo Circunscrito es el que proporciona los puntos de inserción de módulo más convenientes (coincidentes con inicios y fines de intervalo) se plantea un método de resolución en el que se combinan ambos modelos.

Este método consiste en suponer que la poligonal inicial es la "*segmentación*" de una poligonal conjugada y el problema consistiría en determinar dicha poligonal conjugada. Con este algoritmo se conseguiría que los puntos de inserción fuesen aproximadamente los de la poligonal inicial (aprovechando así las ventajas del Algoritmo Circunscrito) y a su vez el tipo de módulo a insertar en cada punto de la *segmentación* (llamada en este caso *segmentación conjugada*) vendría determinado por los incrementos angulares proporcionados por la *poligonal conjugada*, aprovechando así las ventajas del Algoritmo Inscrito.

5.3.3 ALGORITMO DE LA POLIGONAL CONJUGADA.

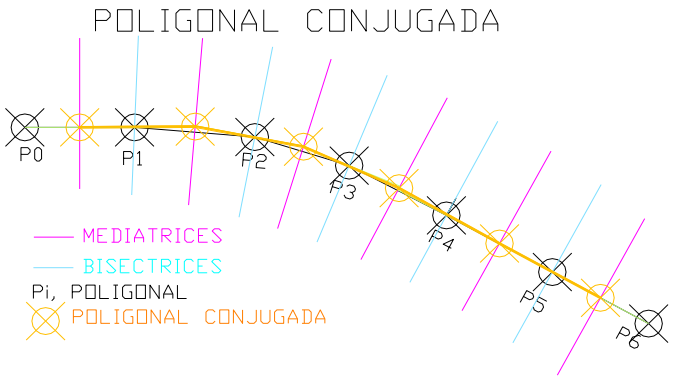
El Algoritmo de la Poligonal Conjugada ofrece buenos resultados bajo cualquier tipo de definición geométrica, si bien su mayor complejidad de cálculo hace preferible su empleo únicamente en situaciones en las que ninguno de los dos algoritmos anteriores (Circunscrito o Inscrito) sea válido (a mayor simplicidad del algoritmo más intuitivo es la depuración de errores y el manejo de datos entre los diversos subsistemas involucrados en la simulación).

El Algoritmo de la Poligonal Conjugada busca las ventajas del Algoritmo Circunscrito, es decir, un número entero de módulos por sección y las ventajas del Algoritmo Inscrito, es decir, una perfecta absorción de variaciones angulares entre módulos adyacentes independientemente del formato empleado en la definición geométrica de la línea directriz (biarco, nube de puntos, NURBS). Con estos objetivos se crea una *poligonal conjugada* que va a representar el mismo papel que la poligonal en el Algoritmo Inscrito.

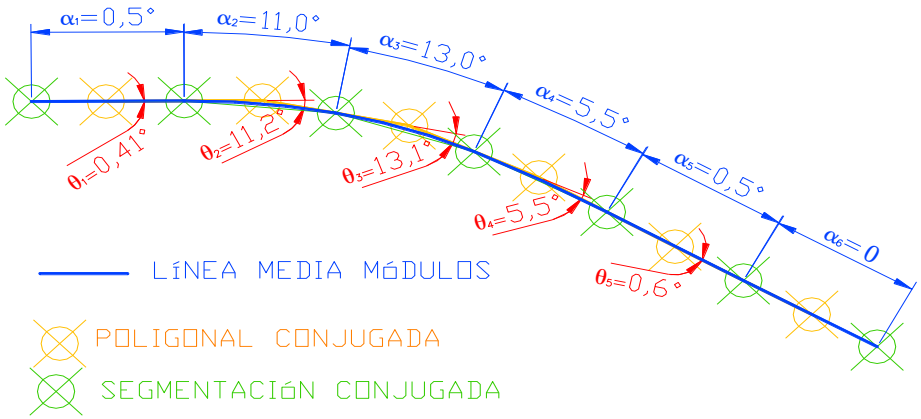
El Algoritmo de la Poligonal Conjugada, al igual que el resto de algoritmos de posicionamiento, parte de la división de la línea directriz en una serie de intervalos. Cada

intervalo tendrá su propia poligonal inicial. La *poligonal conjugada* se calcula a partir de dicha poligonal inicial mediante intersección de las mediatrices y las perpendiculares a las bisectrices de los diferentes segmentos constitutivos de ésta según muestra la Figura 5.31 a.

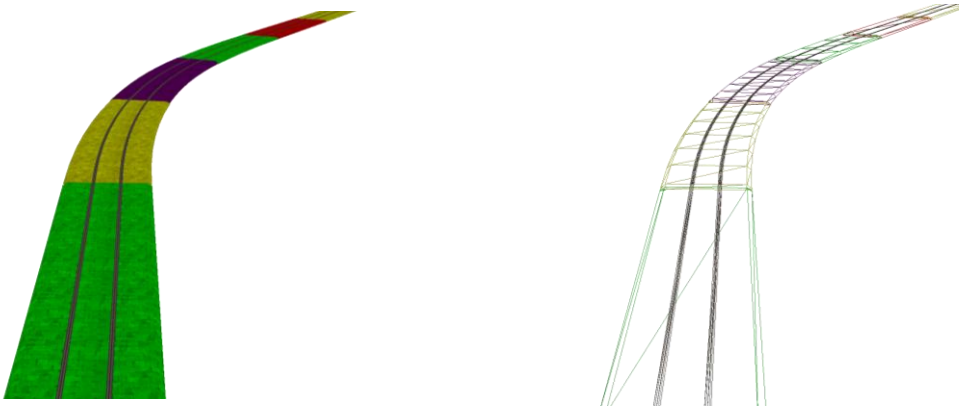
De la misma manera que ocurría con la *segmentación* en el Algoritmo Inscrito, la *segmentación conjugada* será el lugar geométrico de puntos de inserción de módulos. Estos puntos se calculan sobre los segmentos de la *poligonal conjugada* y se distribuyen simétricamente respecto de los vértices de ésta, de manera que absorban sus variaciones angulares sin errores de solapamiento ni aberturas. De este modo los módulos inscriben la *poligonal conjugada* (Figura 5.31 b, c).



(a) Poligonal conjugada.



(b) Segmentación conjugada y línea media de módulos.



(c) Aspecto visual final.

Figura 5.31.Algoritmo de la poligonal conjugada.

Como se comentó en los algoritmos de posicionamiento vistos anteriormente, la longitud de un intervalo de poligonal L', no tiene porque ser múltiplo de la longitud del módulo a emplear (longitud básica, Lb). Como consecuencia, los módulos de longitud básica deberán ser combinados con módulos especiales.

Sean L_i la longitud del segmento P_iP_{i+1} de *poligonal conjugada* y *e* el escalado máximo admisible, entonces:

- Si L_b-*e* ≤ L_i ≤ L_b +*e*, entonces el punto de inserción es el punto medio del segmento de longitud L_i, P_iS_i=(P_iP_{i+1})*0.5. Este módulo absorberá el incremento angular que se produce en el vértice P_{i+1} de la poligonal conjugada.
- Si L_i > L_b +*e*, entonces el segmento L_i dará lugar a dos puntos de inserción de módulo. El primero distará del extremo inicial del segmento una distancia d=L_b*0,5 y el segundo distará del extremo final del segmento la misma distancia. El punto S_i, será el punto de inserción inicial de módulo especial, que cubrirá la longitud (L_i - L_b). El punto S_{i+1} será el punto de inserción de un módulo que absorberá el incremento angular que se produce en el vértice P_{i+1} de la poligonal conjugada. De esta manera los módulos asentados sobre los puntos de la segmentación conjugada seguirán una distribución paralela a los puntos de la poligonal inicial, con la introducción de puntos adicionales en aquellos casos en los que se produzcan longitudes anómalas, evitando nuevamente, los módulos a caballo entre intervalos.
- Si L_i < *Lb-e*, se procederá a la eliminación de un punto de la poligonal conjugada.
- Si L_i ≥ (*Lb-e*)*2 se dividirá el segmento en dos partes iguales.

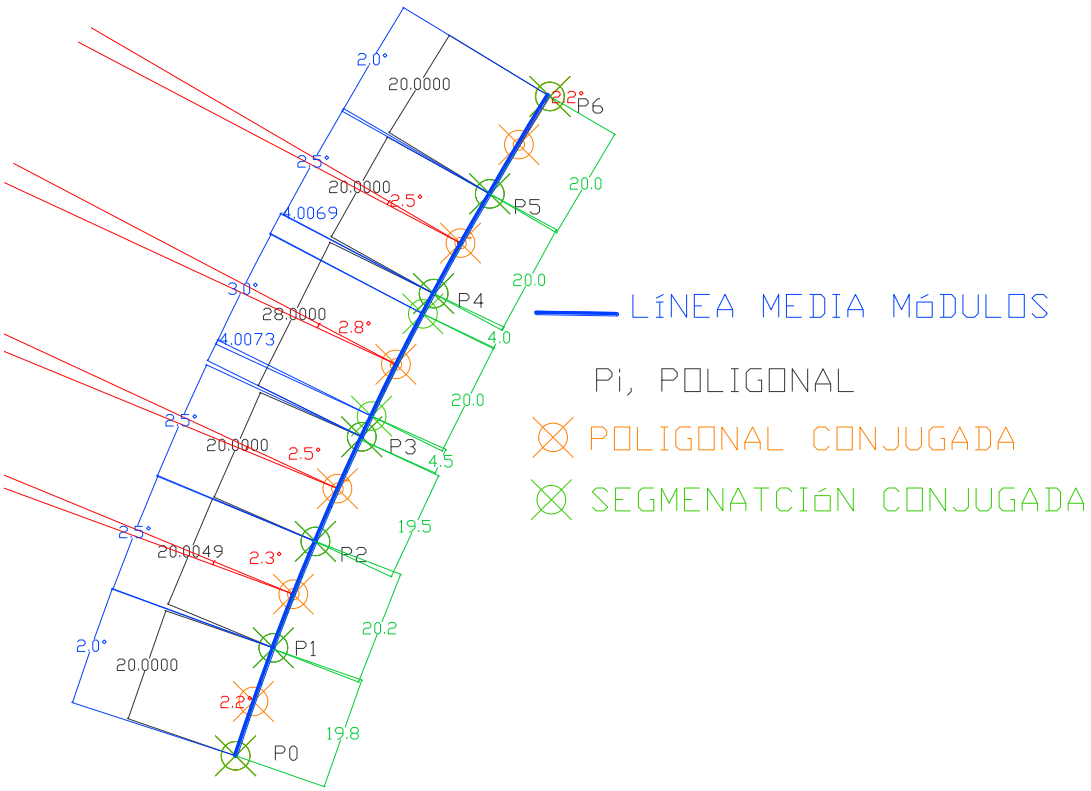


Figura 5.32.Línea media de módulos empleando como módulos especiales módulos de longitudes ad hoc.

5.3.4 COMPARATIVA ENTRE ALGORITMOS.

CONCLUSIONES DEL ALGORITMO CIRCUNSCRITO:

- **Ventajas:** puntos de inserción sobre el biarco de cada intervalo. Por tanto, los intervalos ocupan un número entero de módulos. El único error que se produce es como consecuencia de la discretización angular de los módulos.
- **Inconvenientes**:
  - Solo es válido cuando la definición base es un biarco.
  - El insertar los puntos de cambio de curvatura implica un mayor número de módulos.

CONCLUSIONES DEL ALGORITMO INSCRITO:

- **Ventajas:**
  - Válido para cualquier descripción geométrica base. El único error que se produce es como consecuencia de la discretización angular de los módulos.
  - No es necesario introducir puntos de cambio de curvatura.
- **Inconvenientes:**
  - Módulos a caballo entre intervalos.

CONCLUSIONES ALGORITMO POLIGONAL CONJUGADA:

- **Ventajas**:
  - Las mismas que el algoritmo inscrito y la ventaja adicional de que un intervalo implica un número entero de módulos.
- **Inconvenientes**:
  - En el caso de disponer de un trazado definido por radios, como para la determinación de la poligonal conjugada influyen 4 puntos de la poligonal inicial, la definición geométrica que proporcionan los módulos no es tan precisa como la del Algoritmo Circunscrito, los módulos pueden implicar ciertos desfases respecto de la definición de radios.
  - En el caso de existir radios de longitud pequeña, es posible que la poligonal se "salte" estos radios. Para evitar esto**,** al calcular la poligonal inicial, la discretización del biarco debería incluir los puntos de inicio y fin de arcos cuya longitud sea inferior a la longitud básica, porque sino se pierde en gran medida la descripción geométrica base.
    Por tanto, el caso más indicado para usar la poligonal conjugada es aquél en el que se dispone como definición geométrica base de una nube de puntos o NURBS.

Como conclusión final se establece, que con el fin de hacer que el manejo de la información sea lo más cómodo y sencillo posible para todos los subsistemas involucrados en la simulación, evitando así al máximo posible errores, el orden de prioridades a la hora de generar el trazado será:

- Se intentará siempre emplear como formato de partida el biarco. En el caso de disponer de una nube de puntos se suavizará, siempre y cuando las deformaciones longitudinales y de forma lo permitan y se transformará al formato biarco. En el caso de disponer de biarcos el algoritmo a emplear será el Circunscrito.
- En el caso de disponer como información de partida de una nube de puntos o NURBS, en

el que los puntos de cambio de familia modular sean solo aproximativos se empleará el Algoritmo Inscrito.

- En el caso de disponer como información de partida de una nube de puntos o NURBS y se desee que se respecten con exactitud los puntos de cambio de familia modular se empleará el Algoritmo de la Poligonal Conjugada.

5.4 TRANSFORMACIONES GEOMÉTRICAS DE MÓDULOS: LOS SHADERS.

Un shader es un programa ejecutable en la propia GPU y que puede actuar a diferentes niveles. Existen tres estándares actualmente de lenguaje de shader, HLSL (High Level Shading Language) propiedad de Microsoft para su empleo con Direct3D, GLSL[307] (OpenGL Shading Language) que es el utilizado en esta Tesis al estar desarrollado el motor gráfico sobre OpenSceneGraph basado en OpenGL y por último está CG propiedad del fabricante de aceleradores gráficos Nvidia.

Los shaders [308] permiten la modificación en tiempo real tanto de vértices (*vertex shaders*) como de pixeles (*fragment shaders*), la generación de nueva geometría (*geometry shaders*) y la teselación de la misma en base a un conjunto de bloques (*patches*) definidos por el programador (*tesselation shaders*).

En los desarrollos realizados para los motores gráficos empleados por CITEF los shaders son utilizados para obtener efectos de iluminación, perspectiva y manipulación del grafo de la escena incrementando así las funcionalidades ofrecidas por OpenSceneGraph.

Los shaders que esta Tesis desarrolla, van encaminados a la modificación geométrica del módulo a través de *vertex shaders*. Con estos shaders se reducen el número de geometrías que compone una familia modular y se amplían las posibilidades del mismo.

5.4.1 SHADERS DE TRANSFORMACIÓN SIMPLE.

Como se comentó en el Capítulo 3 la Tecnología Modular parte de un módulo recto, que se curva en planta en tiempo de precarga una serie de grados, constituyendo lo que se denomina un *conjunto base de elementos modulares o módulos.*

Mediante el empleo de shaders, la presente Tesis pretende deformar el módulo recto en tiempo de renderizado suministrando para ello al motor gráfico una serie de parámetros de deformación. Estos parámetros pueden tomar valores continuos cualesquiera, lo que equivaldría a un *conjunto base de infinitos elementos modulares*. Por otro lado, mientras que en tiempo de precarga tan solo se aplica un curvado en planta, el empleo de shaders permite aplicar transformaciones diversas.

En esta Tesis se definen tres deformaciones básicas para el módulo: curvado en XY, curvado en XZ, y curvado en YZ. La Figura 5.33 muestra las vistas ortogonales en alambre y la perspectiva renderizada de un módulo que se ha curvado en la planta.

[307] http://www.opengl.org/documentation/glsl/ [Consulta: 12 Enero 2013].

[308] Wolff, D. OpenGL 4.0 Shading Language Cookbook. Packt Publishing 2011. ISBN: 978-1849514767.

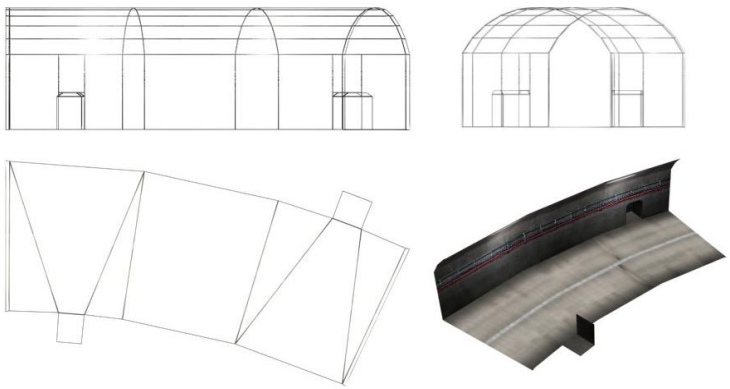


Figura 5.33.Curvado en planta de un módulo.

Sea un sistema de referencia donde el eje X es longitudinal apuntando hacia puntos kilométricos crecientes, Z es el eje vertical hacia arriba e Y es el eje transversal formando un triedro a derechas (Figura 5.34).

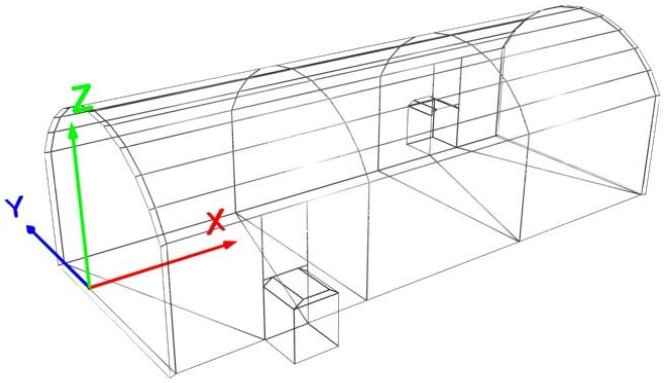


Figura 5.34.Sistema de referencia de un módulo.

La transformación matemática para el curvado en planta de un módulo de longitud inicial L sería la expresada en la ecuación (1), donde x', y', z' son las coordenadas transformadas de un vértice originalmente en x, y, z. Dicha transformación curvaría un ángulo γ y la cuerda de la deformada seguiría siendo L.

$$x' = \frac{L}{2sen(\gamma/2)}\left(sen\left(\gamma\left(\frac{x}{L}-\frac{1}{2}\right)\right)+sen(\gamma/2)\right) \quad y' = \frac{L}{2sen(\gamma/2)}\left(\cos\left(\gamma\left(\frac{x}{L}-\frac{1}{2}\right)\right)-\cos(\gamma/2)\right)$$

$$z' = z$$

Para una mayor eficiencia en la programación de los shaders los términos de funciones trigonométricas se substituyen por aproximaciones polinómicas de las mismas.

Las siguientes deformaciones básicas corresponden al curvado según el eje Y (Figura 5.35), que permite la transición suave entre cambios de pendiente y al curvado entorno al eje X o peraltado del módulo (Figura 5.36).

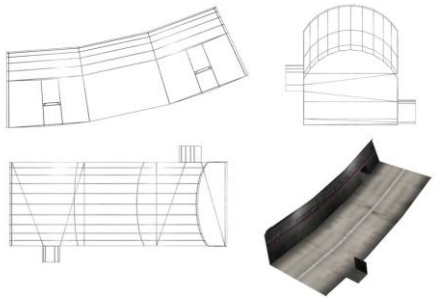


Figura 5.35.Curvado según eje Y un módulo.

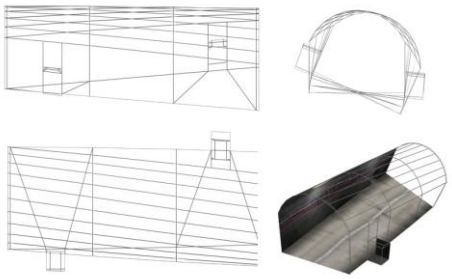


Figura 5.36.Curvado según eje X un módulo.

Las ecuaciones que rigen el curvado en alzado son formalmente iguales a las de planta pero intercambiando *y* con *z*. La ecuación de la torsión es la de un giro alrededor del eje X proporcional al avance en el mismo (2).

$$x' = x \quad y' = y \cdot \cos(\gamma(x/L)) + y \cdot sen(\gamma(x/L)) \quad z' = -y \cdot sen(\gamma(x/L)) + z \cdot \cos(\gamma(x/L))$$

5.4.2 SHADERS DE TRANSFORMACIÓN COMPUESTA TOTAL Y CORREGIDA.

A la hora de representar un entorno virtual para simuladores de conducción terrestre, las normativas constructivas del trazado exigen la combinación de las transformaciones simples vistas. Un aspecto a considerar cuando se combinan deformaciones es que éstas no son conmutativas, el orden de aplicación conduce a resultados distintos y además las transformaciones se acoplan, apareciendo un peralte cuando hay un curvado en planta y en alzado. De esta manera, lo que se aplica es una deformación, no siempre descomponible en combinaciones de las simples, a la que se exige como condiciones de contorno en sus extremos, el perfecto acoplamiento con las secciones transversales de los módulos adyacentes. En la Figura 5.37 se muestra una secuencia de cuatro módulos que siguen una curva que parte de una pendiente y un peralte nulos que van incrementándose. Se puede observar cómo las transiciones entre módulos son imperceptibles, sin holguras.

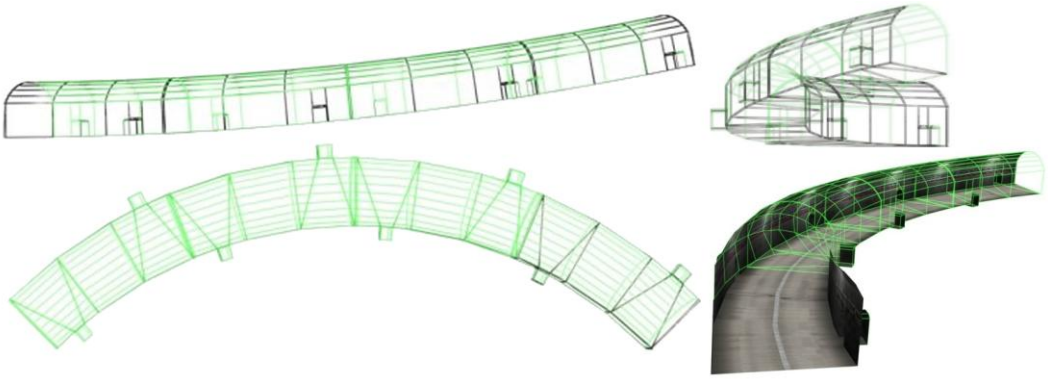


Figura 5.37.Secuencia de varios módulos sometidos a una transformación compuesta.

La Tecnología Modular parte de un módulo recto de longitud básica L_B. Esta longitud es medida en la dirección del eje i_1, según muestra la Figura 5.38, que a su vez coincide con la dirección marcada por la línea directriz que lo posiciona y su sentido es el marcado por los pks crecientes.

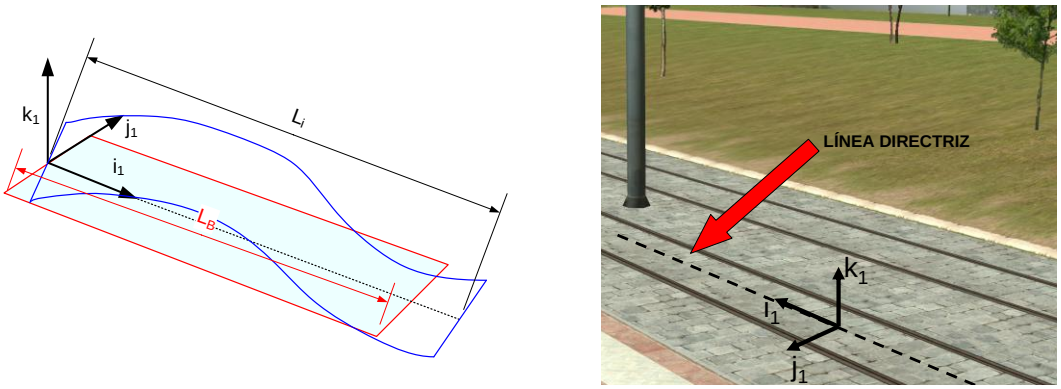


Figura 5.38.Módulo de longitud básica LB empleado para cubrir un tramo de longitud Li.

Mediante el shader de transformación compuesta lo que busca la Tecnología Modular es someter al módulo a una deformación que garantice el acoplamiento perfecto con los módulos adyacentes, tras ser posicionados mediante los algoritmos de Posicionamiento Modular. Para ello lo primero es definir un parámetro adimensional **s**, definido como la relación entre la coordenada $\mathbf{x_0}$ y la longitud básica $\mathbf{L_B}$. El parámetro **s** variará entre 0 y 1.

$$s = \frac{x_0}{L_B}$$
$$0 \le s \le 1$$

Se busca definir una función que garantice un desplazamiento nulo en los extremos del módulo y cuyos ángulos de tangencia, coincidan con los que se desean tener (los definidos por algoritmos de Posicionamiento Modular).

De esta manera, la función que determina el desplazamiento en ***y*** al que han de verse sometidos los puntos del módulo ha de verificar:

$$f_{\Delta y}(s) \equiv \begin{cases} f_{\Delta y}(0) = 0 \\ f_{\Delta y}(1) = 0 \\ \dfrac{df_{\Delta y}}{dx_R}(s=0) = \tan\gamma_{ini}\,(1) \\ \dfrac{df_{\Delta y}}{dx_R}(s=1) = \tan\gamma_{fin} \end{cases}$$

En la ecuación anterior $\mathbf{x_R}$ representa la coordenada real del módulo después de escalarlo. La relación con $\mathbf{x_0}$, coordenada de recorrido del módulo antes de la deformación, y con el espacio recorrido ***s*** es:

$$\begin{aligned} x_R &= x_0 \cdot E_x \\ \frac{dx_R}{dx_0} &= E_x \\ \frac{dx_0}{ds} &= L_B \\ \frac{dx_R}{ds} &= L_B \cdot E_x = L_i \end{aligned} \qquad (2)$$

siendo E_x el escalado longitudinal.

El grupo de ecuaciones (1) se puede volver a reescribir teniendo en cuenta (2) como:

$$f_{\Delta y}(s) \equiv \begin{cases} f_{\Delta y}(0) = 0 \\ f_{\Delta y}(1) = 0 \\ \dfrac{df_{\Delta y}}{dx_R}(s=0) = \dfrac{df_{\Delta y}}{dx_R}\dfrac{ds}{ds}(s=0) = \dfrac{df_{\Delta y}}{ds}\dfrac{ds}{dx_R}(s=0) = \tan\gamma_{ini} \Rightarrow \\ \dfrac{df_{\Delta y}}{ds}(s=0) = L_i \cdot \tan\gamma_{ini} \\ \dfrac{df_{\Delta y}}{ds}(s=1) = L_i \cdot \tan\gamma_{fin} \end{cases} \qquad (3)$$

Las siguientes funciones $\mathbf{f_1}$ y $\mathbf{f_2}$ tienen las siguientes propiedades:

$$\begin{aligned} f_1(s) = s\cdot(1-s)^2 \quad & f_2(s) = s^2\cdot(1-s) \\ f_1(0) = 0 \quad & f_2(0) = 0 \\ f_1(1) = 0 \quad & f_2(1) = 0 \\ \frac{df_1}{ds}(s) = (1-s)(1-3s) \quad & \frac{df_2}{ds}(s) = s\cdot(2-3s) \\ \frac{df_1}{ds}(0) = 1 \quad & \frac{df_2}{ds}(0) = 0 \\ \frac{df_1}{ds}(1) = 0 \quad & \frac{df_2}{ds}(0) = -1 \end{aligned} \qquad (4)$$

Componiendo adecuadamente los requisitos de (3) y los resultados de (4) se tiene una función adecuada para $\mathbf{f_{\Delta y}}$:

$$\begin{aligned} f_{\Delta y}(s) &= f_1(s)\cdot L_i \cdot \tan\gamma_{ini} - f_2(s)\cdot L_i \cdot \tan\gamma_{fin} \\ f_{\Delta y}(0) &= 0 \\ f_{\Delta y}(1) &= 0 \\ \frac{df_{\Delta y}}{ds}(0) &= L_i \cdot \tan\gamma_{ini} \\ \frac{df_{\Delta y}}{ds}(1) &= L_i \cdot \tan\gamma_{fin} \end{aligned} \qquad (5)$$

Para los incrementos en **z**, $\mathbf{f_{\Delta z}}$ habrá que adaptar las condiciones (3) convenientemente:

$$f_{\Delta z}(s) \equiv \begin{cases} f_{\Delta z}(0) = 0 \\ f_{\Delta z}(1) = 0 \\ \dfrac{df_{\Delta z}}{ds}(s=0) = -\dfrac{L_i \tan\beta_{ini}}{\cos\gamma_{ini}} \\ \dfrac{df_{\Delta z}}{ds}(s=1) = -\dfrac{L_i \tan\beta_{fin}}{\cos\gamma_{fin}} \end{cases} \quad (6)$$

Donde el signo negativo indica que el giro β tiene sentido opuesto a la derivada. La función de desplazamiento queda finalmente:

$$f_{\Delta z}(s) = \frac{-f_1(s)\cdot L_i \tan\beta_{ini} + f_2(s)\cdot L_i \tan\beta_{fin}}{\cos\gamma(s)} \quad (7)$$

No hay desplazamientos por deformación en la dirección de x, salvo el escalado.

Los giros **α(s), β(s) y γ(s)** que son coherentes con las funciones de desplazamiento descritas se obtienen derivando. Una reflexión sencilla nos indica que la derivada con respecto a $\mathbf{x_R}$ de la función de desplazamiento en ***y*** se corresponde con el giro **γ(s)**. La derivada del desplazamiento en ***z*** corregida por el **cosγ** proporciona el ángulo **β(s)**. El ángulo de peralte es una extrapolación lineal de los peraltes inicial y final.

$$f_{\Delta y}(s) = f_1(s)\cdot L_i \tan\gamma_{ini} - f_2(s)\cdot L_i \tan\gamma_{fin}$$

$$\frac{df_{\Delta y}}{ds}(s) = (1-s)(1-3s)L_i \tan\gamma_{ini} - s(2-3s)L_i \tan\gamma_{fin}$$

$$\frac{df_{\Delta y}}{dx_R} = \tan(\gamma(s)) = \frac{df_{\Delta y}}{ds}\frac{ds}{dx_R} = (1-s)(1-3s)\tan\gamma_{ini} - s(2-3s)\tan\gamma_{fin} \Rightarrow$$

$$\gamma(s) = \arctan((1-s)(1-3s)\tan\gamma_{ini} - s(2-3s)\tan\gamma_{fin}) \quad (8)$$

$$\gamma(0) = \gamma_{ini} \quad \gamma(1) = \gamma_{fin}$$

Para el ángulo **β(s)**:

$$\frac{df_{\Delta z}}{dx_R} = -\frac{\tan(\beta(s))}{\cos\gamma(s)} = -\frac{df_{\Delta z}}{ds}\frac{ds}{dx_R} = \frac{(1-s)(1-3s)\tan\beta_{ini} - s(2-3s)\tan\beta_{fin}}{\cos\gamma(s)} \Rightarrow$$

$$\beta(s) = \arctan((1-s)(1-3s)\tan\beta_{ini} - s(2-3s)\tan\beta_{fin}) \quad (9)$$

$$\beta(0) = \beta_{ini} \quad \beta(1) = \beta_{fin}$$

Para el ángulo **α(s)** se emplea una interpolación lineal:

$$\alpha(s) = (1-s)\cdot\alpha_{ini} + s\cdot\alpha_{fin} \quad (10)$$

A partir de la definición de los ángulos $\gamma(s), \beta(s), \alpha(s)$ se trata de obtener las coordenadas de un punto que originalmente pertenece a una sección plana en el plano $\mathbf{x_1=0}$, del sistema original $\mathbf{(i_1, j_1, k_1)}$ (Figura 5.39) cuando giran solidariamente los puntos de la sección $\mathbf{P_1(0,y_1^1, z_1^1)}$ y los sistemas de coordenadas [309]. La secuencia de giros es primero $\boldsymbol{\gamma}$ alrededor del eje $\mathbf{k_1}$, luego $\boldsymbol{\beta}$ alrededor del eje $\mathbf{j_2}$ y finalmente un giro $\boldsymbol{\alpha}$ alrededor del eje $\mathbf{i_3}$.

El primer movimiento consiste en un giro $\boldsymbol{\gamma}$ alrededor del eje $\mathbf{k_1}$:

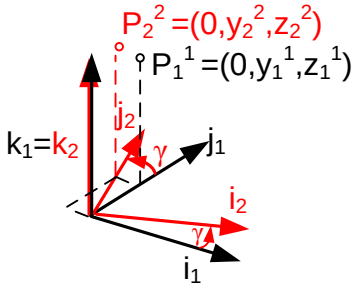


Figura 5.39.Orientación de sección en γ.

Al girar solidariamente el punto $\mathbf{P_1}$ y el sistema de referencia $\mathbf{(i_1, j_1, k_1)}$ es evidente que las coordenadas del punto girado $\mathbf{P_1}$ en el nuevo sistema $\mathbf{(i_2, j_2, k_2)}$ son las mismas, es decir:

$$(0, y_1^1, z_1^1) = (0, y_2^2, z_2^2) \quad (11)$$

Lo que interesa ahora son las coordenadas del punto girado $\mathbf{P_2^1(x_2^1, y_2^1, z_2^1)}$ en el sistema $\mathbf{(i_1, j_1, k_1)}$, para ello se utilizara la matriz $\mathbf{M^{12}}$ de cambio de base de $\mathbf{(i_2, j_2, k_2)}$ a $\mathbf{(i_1, j_1, k_1)}$:

$$M^{12} = M_\gamma \equiv \begin{bmatrix} \cos\gamma & -sen\gamma & 0 \\ sen\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (12)$$

$$P_2^1(x_2^1, y_2^1, z_2^1) = M^{12} \cdot P_2^{2T} \begin{pmatrix} 0 \\ y_2^2 \\ z_2^2 \end{pmatrix}$$

Teniendo en cuenta la ecuación (11) queda:

$$P_2^1(x_2^1, y_2^1, z_2^1) = M^{12} \cdot P_1^{1T} \begin{pmatrix} 0 \\ y_1^1 \\ z_1^1 \end{pmatrix} \quad (13)$$

$$x_2^1 = -y_1^1 sen\gamma$$
$$y_2^1 = y_1^1 \cos\gamma$$
$$z_2^1 = z_1^1$$

El segundo movimiento consiste en un giro $\boldsymbol{\beta}$ alrededor del eje $\mathbf{j_2}$ del punto $\mathbf{P_2}$ solidariamente con el sistema $\mathbf{(i_2, j_2, k_2)}$. Se obtiene un punto $\mathbf{P_3}$ y un sistema $\mathbf{(i_3, j_3, k_3)}$.

[309] Anexo 4.

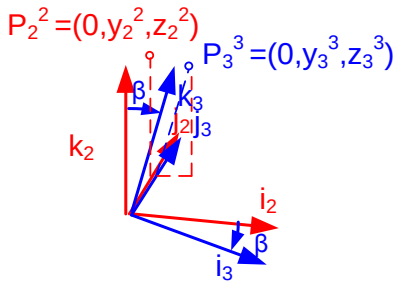


Figura 5.40.Giro de pendiente de la sección.

La ecuación equivalente a la (11) es la siguiente:

$$(0, y_3^3, z_3^3) = (0, y_2^2, z_2^2) = (0, y_1^1, z_1^1) \quad (14)$$

Interesa conocer las coordenadas de $\mathbf{P_3}$ en el sistema **1**, es decir, $\mathbf{P_3^1(x_3^1, y_3^1, z_3^1)}$. Para ello es necesario obtener primero $\mathbf{P_3}$ en el sistema **2**, es decir, $\mathbf{P_3^2(x_3^2, y_3^2, z_3^2)}$:

$$M^{23} = M_\beta \equiv \begin{bmatrix} \cos\beta & 0 & sen\beta \\ 0 & 1 & 0 \\ -sen\beta & 0 & \cos\beta \end{bmatrix}$$

$$P_3^2(x_3^2, y_3^2, z_3^2) = M^{23} \cdot P_3^{3T} \begin{pmatrix} 0 \\ y_3^3 \\ z_3^3 \end{pmatrix} = M^{23} \cdot P_1^{1T} \begin{pmatrix} 0 \\ y_1^1 \\ z_1^1 \end{pmatrix}$$

$$\begin{aligned} x_3^2 &= z_1^1 sen\beta \\ y_3^2 &= y_1^1 \\ z_3^2 &= z_1^1 \cos\beta \end{aligned} \quad (15)$$

Luego vendría calcular $\mathbf{P_3^1(x_3^1, y_3^1, z_3^1)}$ en función de $\mathbf{P_3^2(x_3^2, y_3^2, z_3^2)}$ siguiendo el mismo proceso descrito en (13), pero esta vez para el punto $\mathbf{P_3}$ en vez del $\mathbf{P_2}$:

$$P_3^1(x_3^1, y_3^1, z_3^1) = M^{12} \cdot P_3^{2T} \begin{pmatrix} x_3^2 \\ y_3^2 \\ z_3^2 \end{pmatrix}$$

$$\begin{aligned} x_3^1 &= x_3^2 \cos\gamma - y_3^2 sen\gamma \\ y_3^1 &= x_3^2 sen\gamma + y_3^2 \cos\gamma \\ z_3^1 &= z_3^2 \end{aligned} \quad (16)$$

Si se desarrolla (16) sustituyendo (15) se tiene:

$$P_3^1(x_3^1, y_3^1, z_3^1) = M^{12} \cdot P_3^{2T} \begin{pmatrix} x_3^2 \\ y_3^2 \\ z_3^2 \end{pmatrix} = M^{12} M^{23} \cdot P_3^{3T} \begin{pmatrix} 0 \\ y_3^3 \\ z_3^3 \end{pmatrix}$$

$$= M^{12} M^{23} \cdot P_1^{1T} \begin{pmatrix} 0 \\ y_1^1 \\ z_1^1 \end{pmatrix}$$

$$x_3^1 = z_1^1 sen\beta \cos\gamma - y_1^1 sen\gamma \qquad (17)$$

$$y_3^1 = z_1^1 sen\beta \cdot sen\gamma + y_1^1 \cos\gamma$$

$$z_3^1 = z_1^1 \cos\beta$$

La matriz para pasar directamente de 3 a 1 sería $\mathbf{M^{13} = M^{12} \cdot M^{23}}$:

$$M^{13} = M^{12} M^{23} = M_\gamma M_\beta \equiv$$

$$\begin{bmatrix} \cos\gamma & -sen\gamma & 0 \\ sen\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\beta & 0 & sen\beta \\ 0 & 1 & 0 \\ -sen\beta & 0 & \cos\beta \end{bmatrix} = (18)$$

$$\begin{bmatrix} \cos\gamma\cos\beta & -sen\gamma & \cos\gamma \cdot sen\beta \\ sen\gamma\cos\beta & \cos\gamma & sen\gamma \cdot sen\beta \\ -sen\beta & 0 & \cos\beta \end{bmatrix}$$

El tercer y último movimiento consiste en un giro **α** alrededor del eje $\mathbf{i_3}$ del punto $\mathbf{P_3}$ solidariamente con el sistema $\mathbf{(i_3, j_3, k_3)}$. Se obtiene un punto $\mathbf{P_4}$ y un sistema $\mathbf{(i_4, j_4, k_4)}$.

La ecuación equivalente a la (14) es la siguiente:

$$(0, y_4^4, z_4^4) = (0, y_3^3, z_3^3) = (0, y_2^2, z_2^2) = (0, y_1^1, z_1^1) \quad (19)$$

Interesa conocer las coordenadas de $\mathbf{P_4}$ en el sistema **1**, es decir, $\mathbf{P_4^1(x_4^1, y_4^1, z_4^1)}$. Para ello es necesario obtener primero $\mathbf{P_4}$ en el sistema **3**, es decir, $\mathbf{P_4^3(x_4^3, y_4^3, z_4^3)}$. Posteriormente se halla $\mathbf{P_4^2(x_4^2, y_4^2, z_4^2)}$ y finalmente $\mathbf{P_4^1(x_4^1, y_4^1, z_4^1)}$.

Para obtener $\mathbf{P_4^3(x_4^3, y_4^3, z_4^3)}$ hay que aplicar $\mathbf{M^{34}}$:

$$M^{34} = M_\alpha \equiv \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -sen\alpha \\ 0 & sen\alpha & \cos\alpha \end{bmatrix}$$

$$P_4^3(x_4^3, y_4^3, z_4^3) = M^{34} \cdot P_4^{4T} \begin{pmatrix} 0 \\ y_4^4 \\ z_4^4 \end{pmatrix} = M^{34} \cdot P_1^{1T} \begin{pmatrix} 0 \\ y_1^1 \\ z_1^1 \end{pmatrix} \quad (20)$$

Se obtiene las coordenadas intermedias:

$$P_4^3(x_4^3, y_4^3, z_4^3) = M^{34} \cdot P_1^{1T} \begin{pmatrix} 0 \\ y_1^1 \\ z_1^1 \end{pmatrix}$$

$$x_4^3 = 0$$
$$y_4^3 = y_1^1 \cos\alpha - z_1^1 \cdot sen\,\alpha \quad (21)$$
$$z_4^3 = y_1^1 sen\,\alpha + z_1^1 \cdot \cos\alpha$$

A partir de $\mathbf{P_4^3(0,\ y_4^3,\ z_4^3)}$ se puede obtener directamente $\mathbf{P_4^1(x_4^1,\ y_4^1,\ z_4^1)}$ por la aplicación directa de la matriz de cambio de 3 a 1 $\mathbf{M^{13}}$ descrita en (18):

$$x_4^1 = z_4^3 sen\beta \cos\gamma - y_4^3 sen\gamma$$
$$y_4^1 = z_4^3 sen\beta \cdot sen\gamma + y_4^3 \cos\gamma \quad (22)$$
$$z_4^1 = z_4^3 \cos\beta$$

Si se sustituye $\mathbf{y_4^3,\ z_4^3}$ de (21) en (22) se obtiene las coordenadas del punto transformado por los tres giros en el sistema inicial en función de las coordenadas iniciales:

$$x_4^1 = y_1^1 sen\alpha \cdot sen\beta \cdot \cos\gamma + z_1^1 \cdot \cos\alpha \cdot sen\beta \cdot \cos\gamma - y_1^1 \cos\alpha \cdot sen\gamma + z_1^1 \cdot sen\alpha \cdot sen\gamma$$
$$y_4^1 = y_1^1 sen\alpha \cdot sen\beta \cdot sen\gamma + z_1^1 \cdot \cos\alpha \cdot sen\beta \cdot sen\gamma + y_1^1 \cos\alpha \cos\gamma - z_1^1 \cdot sen\alpha \cos\gamma \quad (23)$$
$$z_4^1 = y_1^1 sen\alpha \cdot \cos\beta + z_1^1 \cdot \cos\alpha \cdot \cos\beta$$

5.5 JERARQUIZACIÓN ESCÉNICA. PARÁMETROS OPTIMIZABLES DE LA TECNOLOGÍA MODULAR.

La Tecnología Modular es responsable de la creación de un grafo de la escena cuya misión es doble: por un lado optimiza la representación gráfica del escenario garantizando en todo momento una velocidad de refresco adecuada y por otro lado aporta la funcionalidad necesaria al mismo.

En cuanto a la optimización de la representación gráfica de la escena, si bien la instanciación supone un enorme ahorro de recursos, esto no es suficiente cuando se trata de simulaciones de conducción terrestre en las que se representan entornos virtuales de gran tamaño. Es necesaria una óptima organización del grafo de la escena que permita decidir en cada momento qué parte del escenario se va a visualizar y con qué nivel de detalle (LOD). En dicho proceso de optimización la Tecnología Modular definirá el valor de diversos parámetros (número de instancias, agrupación de las mismas..etc) cómo se verá a continuación.

En cuanto a la funcionalidad del escenario, existen diversos comportamientos a simular, como la regulación semafórica y el movimiento de agujas. Mediante el empleo de determinados nodos del grafo de la escena, la Tecnología Modular definirá dichos comportamientos.

El grafo de la escena se describe en un archivo de texto plano empleando un Lenguaje de Macros [310] creado en el departamento de Ingeniería Gráfica y Simulación para CITEF. Este lenguaje será el empleado por esta Tesis para la jerarquización del escenario.

5.5.1 EL ESCENARIO.

El escenario virtual está formado por diversos tipos de objetos, cada uno con sus propiedades de representación, sus propiedades físicas, sus leyes de comportamiento y su localización tridimensional.

Los elementos que componen un escenario se clasifican en:

- **Geometrías**

Son los elementos 3d que dotan al escenario de la apariencia visual. Están constituidos por polígonos y su unión constituye los objetos. Como es obvio cada polígono está formado por vértices y líneas, pero además presentan una serie de propiedades que dotan de mayor realismo a los mundos como son:

  - color de aplicación;
  - materiales: determinan el efecto que provoca la luz cuando actúa sobre los polígonos produciendo diferentes efectos y así que cada punto de estos presente una u otra intensidad de color;
  - textura aplicada y forma de aplicación, etc.

- **Efectos atmosféricos**

Permiten definir efectos de ambiente especiales, tales como lluvia, humo, niebla, bruma, etc. El efecto niebla, además de simular el fenómeno atmosférico, permite la reducción de recursos al no calcular el color de todos los pixels de los diferentes objetos, asignando un color específico a los mismos a partir de una determinada distancia. Este tipo de nodo puede afectar a una, a un grupo o a todas las geometrías presentes en el entorno. Además cuando se aplica a gran distancia, minimiza el efecto producido por el plano de yon.

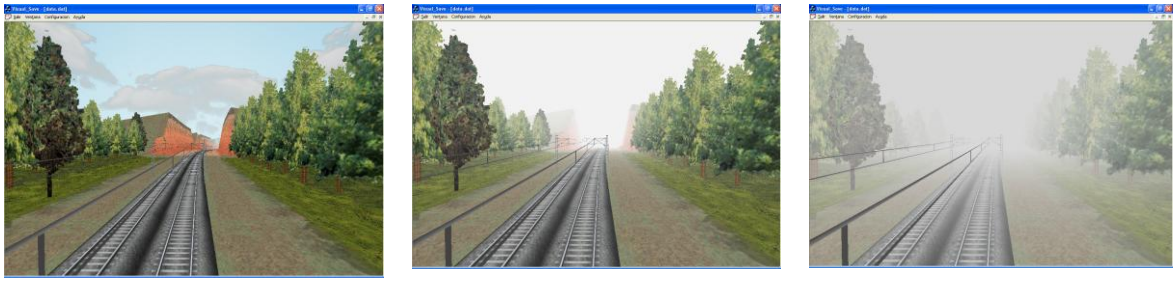


Figura 5.41.Efectos atmosféricos

- **Luces**

Su principal función es la de iluminar los objetos del mundo. Estos elementos presentan una gran importancia, ya que de ellos va a depender en gran medida el consumo de recursos. Un

[310] Anexo 5.

entorno con un gran número de luces necesitará un mayor número de cálculos para determinar el color de cada píxel.

- **Fuentes de sonido**

Según el tipo de hardware que se disponga, se puede simular sonido tridimensional, efecto doppler, etc.

- **Información posicional**

Incluye las coordenadas que permiten definir la localización espacial de cada objeto del escenario. La localización espacial se define mediante las tres coordenadas de un punto de referencia del objeto y mediante los parámetros que definen su orientación en el espacio.

5.5.2 TIPOS DE NODOS EMPLEADOS POR LA TECNOLOGÍA MODULAR.

Los objetos que componen un escenario se codifican en el grafo de la escena mediante el empleo de diversos tipos de nodos. El orden de los nodos en el grafo de la escena es fundamental ya que marca el orden de renderizado (Figura 5.42). El algoritmo de visualización recorre esta estructura de forma recursiva a partir del nodo raíz y selecciona cuáles son los datos enviados al sistema de procesamiento gráfico.

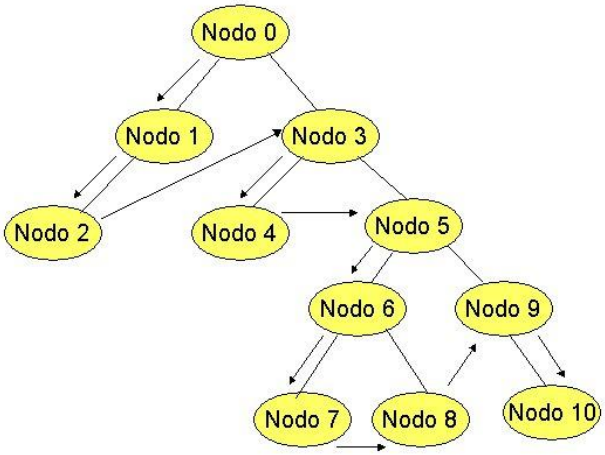


Figura 5.42.Estructura de un scene graph mostrando orden de evaluación.

Los nodos del grafo de la escena contienen información de diverso tipo y ejercen diferentes funciones. Los nodos que maneja la Tecnología Modular para optimizar el escenario son los siguientes:

- **Nodos de geometría**: Como su nombre indica contienen la información de la geometría de la escena. Cuando el algoritmo de recorrido llega a un nodo de geometría (normalmente los nodos terminales del grafo), sus datos son enviados al sistema gráfico y los polígonos correspondientes son dibujados.
- **Nodos de grupo**: Tienen la misión de agrupar varias nodos permitiendo definir operaciones o propiedades sobre estos conjuntos. Esta estructuración agiliza las operaciones de edición de los escenarios y a su vez, posibilita la transferencia de propiedades de unos nodos a otros. Sin embargo, el empleo de estos grupos aumenta el número de cálculos, ya que cada nodo hijo debe expresar su posicionamiento y orientación en coordenadas relativas al grupo. Esto da lugar a que

dicho posicionamiento se calcule por el producto de las matrices de transformación asociadas a todos los nodos que son ancestros del mismo. Como consecuencia de todo esto el número de grupos óptimo será una solución de compromiso entre el aumento de carga computacional y la mayor facilidad de edición.

- **Nodos de nivel de detalle (LOD)**: Estos nodos se utilizan para implementar la selección de detalle. Un nodo LOD forzará al algoritmo a efectuar una comprobación que escogerá uno solo de sus hijos para ser recorrido. Cada hijo representa el mismo objeto pero con un diferente nivel de detalle. Para efectuar la comprobación de distancia estos nodos necesitan almacenar cierta información adicional:
    - Centro: el punto desde el cual se mide la distancia del objeto al observador.
    - Rangos de distancia para cada hijo, es decir, entre qué distancias es visible cada uno de los niveles de detalle. Normalmente se da una lista ordenada de valores de distancia, para los cuales se cambia de un hijo a otro.

  El empleo de LODs dependerá de la complejidad de las geometrías. Si el entorno es muy abierto y presenta una gran dificultad de renderizado puede ser aconsejable su empleo. Sin embargo, no se debe olvidar que cada uno de los hijos del LOD ocupa memoria y lo que en principio puede resultar beneficioso, puede convertirse en un lastre. Por otro lado, los LODs presentan una distancia máxima a partir de la cual el objeto no se visualizará. Aunque esto disminuirá el tiempo de renderizado, el realismo puede verse también disminuido. Las distancias de corte toman por tanto una gran relevancia y de ellas dependerá en gran medida el mayor o menor consumo de recursos, así como el mayor o menor grado de realismo.

  Como conclusión el empleo de LODs (que no sean del tipo "se ve o no se ve", ya que éstos no implican la adición de nuevas geometrías puesto que el segundo nivel de detalle está vacío) traerá consigo un mayor tiempo de carga y un mayor consumo de memoria, al emplearse un mayor número de geometrías y texturas, aunque a cambio se consigue generalmente una mejora de rendimiento.
- **Nodos de control por programa (nodos switch)**: permiten alternar entre diversas representaciones de un objeto. Cada representación vendrá identificada por un índice, de manera que el nodo switch recibirá en tiempo de ejecución el valor del índice del hijo que deberá ser recorrido. Este tipo de nodos se emplean para la simulación del comportamiento de diversos elementos del escenario como los semáforos y agujas.

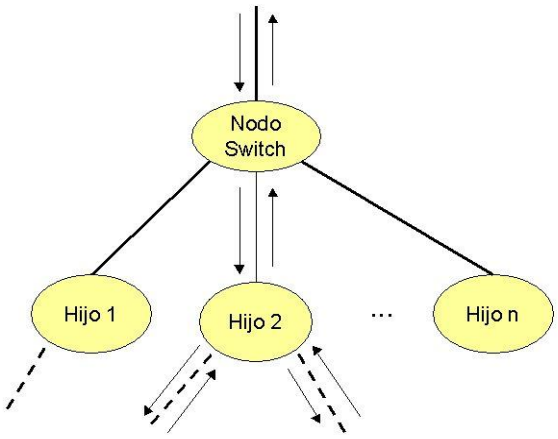


Figura 5.43.Funcionamiento de un nodo switch.

5.5.3 OPTIMIZACIÓN JERARQUICA REALIZADA POR LA TECNOLOGÍA MODULAR.

En este apartado se describirán aquellos parámetros que la Tecnología Modular controla al optimizar un entorno virtual [311]. La vertiginosa evolución del hardware da lugar a que los máximos umbrales admisibles para cada uno de estos parámetros estén en continua mejora. Sin embargo, las altas exigencias de CPU que en una simulación de conducción terrestre existen para otros fines diversos a la representación gráfica, hacen conveniente trabajar siempre lo más lejos posible de dichos umbrales máximos.

5.5.3.1 NÚMERO ÓPTIMO DE MÓDULOS

Para determinar el número idóneo de módulos a emplear en la generación de un determinado entorno virtual es necesario llegar a una solución de compromiso en el cumplimiento de diversos requisitos:

- Tamaño de módulo adecuado para representar cualquier descripción geométrica base.
- Tiempos de carga y memoria de almacenamiento mínimos.
- Grafo de la escena óptimo que facilite la localización y edición de elementos.

Un tamaño de módulo óptimo implica elegir una longitud básica acorde a la longitud media del arco a representar ya que el módulo es un arco de radio constante. Es decir, no tiene ningún sentido emplear una longitud básica de Lb metros, si existe un número elevado de arcos de longitud inferior a dicha magnitud.

En los entornos ferroviarios, urbanos e interurbanos que es necesario representar en las simulaciones de conducción terrestre del CITEF, el tamaño máximo de módulo suele ser de 20 metros. Esta longitud permite representar óptimamente todos los trazados geométricos típicos de estos escenarios, consiguiéndose las reducciones en memoria y tiempo de carga buscadas. Por ejemplo, en la representación de la Línea 7 de Metro de Madrid, los 23 km de túnel han sido generados mediante la creación de 5 familias de túnel de longitud básica 20 metros, obteniéndose los resultados que muestra la siguiente tabla.

Tabla 1. Comparación de los tiempos de carga y memoria empleada en la representación de los túneles de la Línea 7 de Metro de Madrid con y sin el empleo de la Tecnología Modular.

EMPLEANDO TECNOLOGÍA MODULAR	TIEMPOS DE CARGA (segundos)	MEMORIA
SI	3	1.3MB
NO	22	44MB

Es necesario tener en cuenta que si bien el empleo de longitudes de módulo inferiores a esta longitud máxima supone un descenso de los tiempos de carga y de la memoria de

[311] NOTA: Todos los resultados de las pruebas presentadas en este Capítulo han sido llevadas a cabo en un Intel (R) Xeon (TM) Doble Procesador de 3.06 GHz 2GB RAM con tarjeta gráfica NVIDIA 7800 GS. La importancia de dichos resultados no radica tanto en la magnitud absoluta de dichos valores como en los cambios relativos que para un determinado ordenador puede suponer el empleo o no de la Tecnología Modular en la representación de escenarios virtuales simulaciones de conducción terrestre.

almacenamiento necesaria, también puede implicar una serie de efectos negativos: un excesivo número de instancias complica el recorrido del grafo de la escena, implicando más cálculos y un descenso de la velocidad de renderizado. La Figura 5.44 muestra la velocidad de renderizado (*frame rate*) obtenida al representar virtualmente un entorno constituido por una línea ferroviaria modelada a partir de una familia de túnel. Con el fin de analizar la influencia del tamaño del módulo se han generado cinco escenarios distintos, con tamaños de módulo de 1, 3, 5, 10 y 20 metros respectivamente.

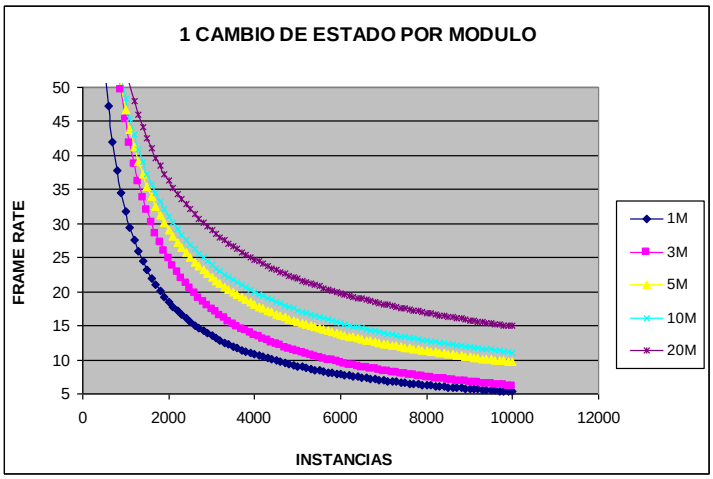


Figura 5.44.Variación de la velocidad de renderizado en función del número de instancias para un escenario generado con diversos tamaños de módulo.

Se puede observar como por ejemplo, la velocidad de renderizado que puede alcanzarse en la visualización de 2000 instancias puede variar desde los 18 fr/s, si el entorno se ha construido con módulos de 1 metro, hasta los 37 fr/s, si se emplean módulos de 20 metros.

Teniendo en cuenta que la velocidad de renderizado objetivo es 30fr/s (por encima de esta velocidad el ojo humano no es capaz de percibir mejoras) el tamaño de LOD elegido determinará la longitud de entorno a visualizar en cada instante y el número de instancias a manejar, lo que vendrá condicionado por la longitud base de módulo elegida. Para el ejemplo comentado previamente, para una distancia de LOD de 2km, una longitud base de 5 metros proporcionaría velocidades superiores a la objetivo.

Sin embargo, para determinar la longitud básica óptima hay que tener en cuenta que ésta varía con el número de cambios de estado (*statesets*) por módulo (Figura 5.45).

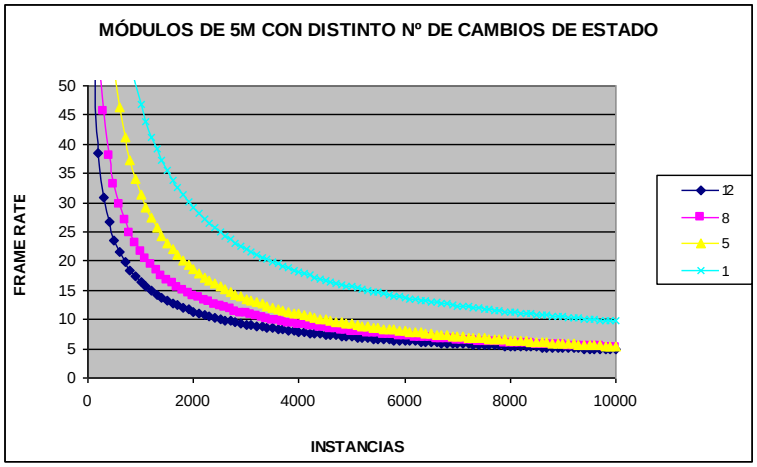


Figura 5.45.Evaluación de la variación de la velocidad de refresco con el número de instancias vistas para un entorno generado con módulos de 5m y distinto número de cambios de estado.

A medida que el número de cambios de estado por módulo aumenta, el máximo número de instancias admisible disminuye. La siguiente figura muestra la variación del máximo número de instancias admisibles en función del número de cambios de estado empleado por módulo.

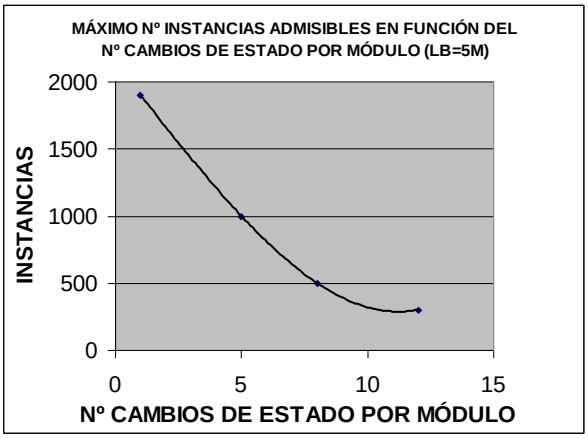


Figura 5.46.Variación del máximo nº de instancias admisibles con el nº de cambios de estado por módulo.

Teniendo en cuenta estos resultados, las familias modulares se diseñarán agrupando las máximas características repetibles posibles de manera que se minimicen el número cambios de estado por módulo (empleando mapas de textura) y el tamaño base de módulo será el mínimo que proporcione una velocidad de renderizado superior a 30 fr/s. Para ello se seguirá un proceso iterativo empleando los diagramas velocidad renderizado/nº instancias y velocidad renderizado/nº cambios de estado vistos (Figura 5.44, Figura 5.45, Figura 5.46). Por tanto, para la línea ferroviaria empleada como ejemplo, la creación de un grafo de la escena con LODs de tamaño 2 km, la longitud básica óptima, teniendo en cuenta que los módulos empleados constan de un único cambio de estado sería de 5 metros.

5.5.3.2POLÍGONOS DEL MÓDULO

La influencia del número de polígonos en la velocidad de renderizado es cada vez menos importante. Su número no debe superar una franja dependiente de dos factores fundamentales: la capacidad del procesador y la de la tarjeta gráfica. Para ello la Tecnología Modular crea geometrías optimizadas y emplea nodos de tipo LOD, que adecuen en cada momento el número de polígonos a representar.

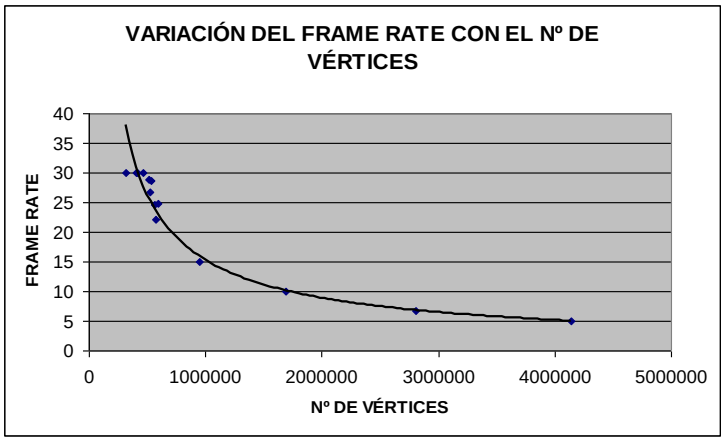


Figura 5.47.Evaluación de la variación de la velocidad de refresco con el número de vértices.

Para la línea ferroviaria tomada como ejemplo se han realizado diversas pruebas empleando módulos de longitud 5m con diferente número de vértices. Los resultados mostrados en la Figura 5.47 revelan cómo siempre y cuando el número de vértices se mantenga inferior a 500.000 la velocidad de renderizado no desciende del valor objetivo (30 fr/s).

A continuación se muestran entornos virtuales creados por la Tecnología Modular y el número de vértices empleado en cada caso. La Figura 5.48 muestra un escenario de túnel en el que se visualizan 6.505 vértices, distribuidos en 108 módulos. En la Figura 5.49, se representa un escenario urbano, con 83.332 vértices. Ambos ejemplos muestran estar muy alejados del umbral máximo de vértices.

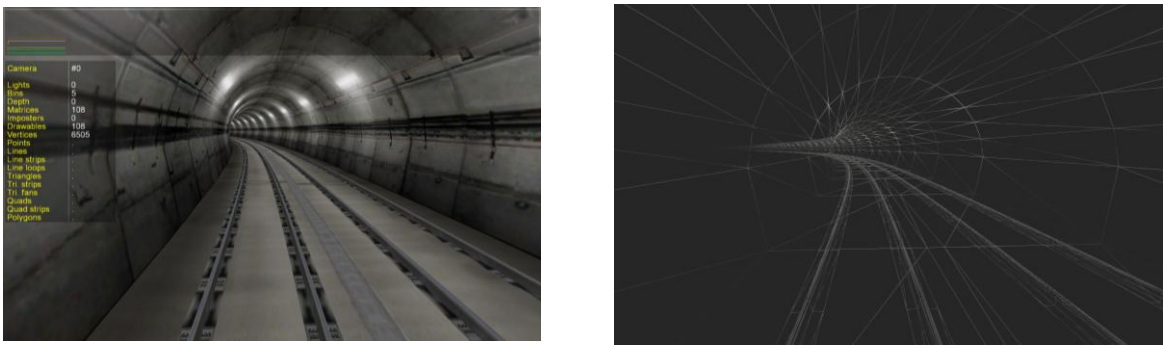


Figura 5.48.Número de vértices empleados en una simulación ferroviaria.

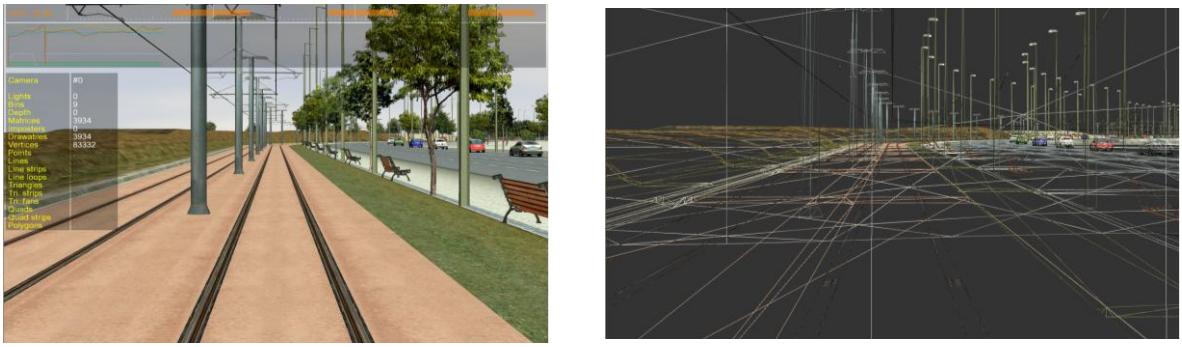


Figura 5.49.Número de vértices empleados en una simulación metropolitana.

5.5.3.3 DISEÑO DE LAS TEXTURAS DEL MÓDULO

La utilización de texturas permite reducir el número de polígonos de las mallas reduciéndose considerablemente el tiempo de representación de las geometrías. Gracias a las texturas se consiguen efectos que sólo podrían ser alcanzados aumentando el número de polígonos y aplicando diferentes materiales a cada uno de ellos, proceso altamente costoso tanto en realización, como en ejecución.

Los estudios llevados a cabo en el departamento de Ingeniería Gráfica y Simulación [312] han concluido que el rendimiento asociado a las texturas está estrechamente relacionado con tres factores fundamentales:

[312] Maroto, J. Metodología para la generación de entornos virtuales distribuidos y su aplicación a simuladores de conducción. Tesis (Doctoral), E.T.S.I. Industrial (UPM).Septiembre 2005.

- **La memoria de la tarjeta gráfica**. Si la memoria ocupada por las texturas supera la de la tarjeta provocará un comportamiento anómalo que se traduce en una disminución de la velocidad de redibujado en determinadas situaciones. Esta caída se debe a que cada vez que tenga que representar un polígono que haga uso de una determinada textura, la librería gráfica comprueba si esta textura está cargada dentro de la memoria de textura, de tal manera que si ésta no se encuentra se producirá una transferencia de información entre la memoria del ordenador y la memoria de la tarjeta, provocando una caída de los recursos. Las bajadas de rendimiento son de un grandísimo nivel. En pruebas realizadas por el Grupo de Ingeniería Gráfica de la U.P.M. para Metro de Madrid, se observó que con sólo sobrepasar en un MByte la memoria de texturas de la tarjeta gráfica el índice de redibujado pasaba de 30 fotogramas por segundo a 6.
- **La presencia de filtros**. Otro de los factores que influye en el número de fotogramas por segundo es la presencia de filtros de texturas. Si el tamaño de un polígono ocupa en pantalla el mismo número de pixels que el tamaño de la textura asociada, entonces el mapeo será un punto de la textura por píxel. Ahora bien esta situación ideal no se suele dar. El tamaño del polígono a representar será menor o mayor que la textura, dando lugar al uso de filtros de ampliación o reducción. Cada filtro implica el consumo de unos determinados recursos que será necesario minimizar.
- **Texturas de potencia 2.** En OpenGL las dimensiones de las texturas deben ser potencias de 2. Por lo tanto, aunque la librería gráfica empleada admite imágenes con resoluciones que no cumplan esta restricción, al estar basada en OpenGL, realiza una conversión interna antes de enviar la imagen a la tarjeta gráfica, lo que trae consigo un mayor consumo de tiempo, que se traduce en una bajada de rendimiento. Por todo ello es aconsejable el tratamiento de la imagen por medio de un programa de edición de las mismas para que presente un alto y ancho potencia de 2. Esta operación no afectará al mapeo de la textura sobre la geometría, ya que cada vértice que constituye la misma se relaciona con la imagen por medio de coordenadas de textura.
- **La presencia de canal alpha**. Gracias al empleo de este canal es posible controlar el grado de opacidad de los polígonos (Figura 5.50). La existencia de semitransparencias complica los cálculos a la hora de determinar las partes vistas y ocultas y también puede ir acompañado de problemas en la visualización, por lo que su empleo debe minimizarse.

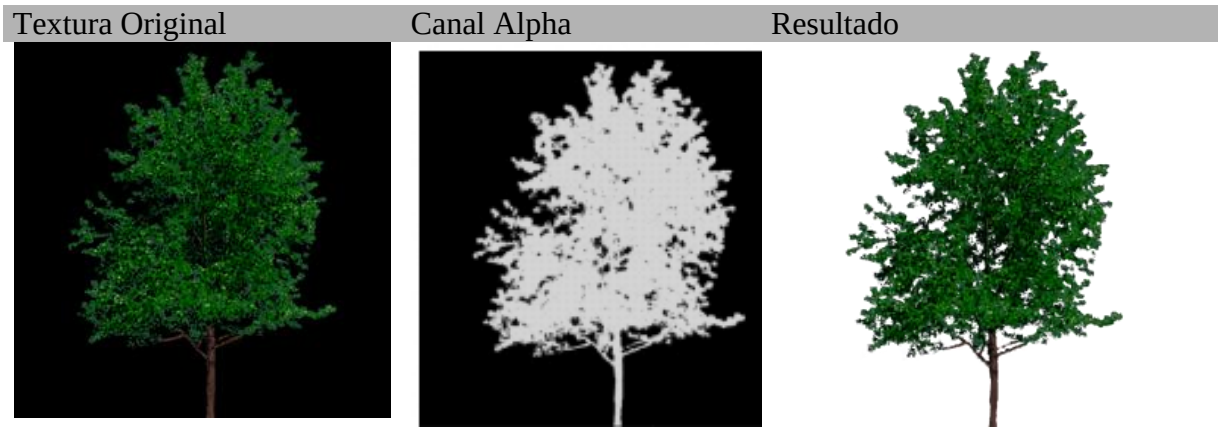


Figura 5.50.Utilización de canal alpha.

5.5.3.4 EMPLEO DE SHADERS

Gracias al empleo de shaders, la Tecnología Modular minimiza los tiempos de carga y la memoria de almacenamiento e incrementa la flexibilidad constructiva del entorno, ya que los

módulos adquieren su aspecto geométrico final mediante deformación en tiempo real, por lo que en tiempo de precarga tan solo es necesario almacenar un módulo recto por cada familia modular.

Las ventajas aportadas por la Tecnología Modular se hacen cada vez más patentes a medida que aumenta el tamaño de los escenarios. La siguiente tabla muestra, para el caso particular de la Línea 7 de Metro de Madrid, los elementos más característicos que se representan en una línea ferroviaria, junto a sus tiempos de carga y la memoria de almacenamiento empleados. La velocidad de renderizado es, en cualquiera de los métodos de generación mostrados, muy superior a la velocidad objetivo (30fr/seg).

Tabla 2. Tiempos de carga y memoria de almacenamiento para Línea 7 de Metro de Madrid.

ELEMENTO	**TIEMPO CARGA L7(seg)**	**MEMORIA TOTAL L7**
SIN SHADERS		
AGUJAS	1.2899167	704KB
SOLAPAS_ESPADINES	1.6662966	1.4MB
VIA	2.1209848	2MB
TUNELES(5FAMILIAS)	3.0123236	1.31MB
CON SHADERS		
AGUJAS	0.4490163	96KB
SOLAPAS_ESPADINES	1.003947	256KB
VIA	0.871326	128KB
TUNELES	0.9941645	384KB
BLOQUES (SIN INSTANCIACIÓN)		
TUNEL_RESOLUCION_3M	22.344373	44MB
CAJAS DE TÚNEL	30.4799973	55MB
ESTACIONES	36.932982	44MB

A partir de los múltiples entornos que la Tecnología Modular ha generado para diferentes proyectos ferroviarios, se han estimado los siguientes tiempos y memorias de almacenamiento promedios:

Tabla 3.Tiempos de carga y memoria de almacenamiento promedio estimados para una línea ferroviaria.

ELEMENTO	**TIEMPO CARGA PROMEDIO (POR MÓDULO O POR KM)**	**MEMORIA PROMEDIO (POR MÓDULO O POR KM)**
SIN SHADERS		
AGUJAS	0.058632577	32KB
SOLAPAS_ESPADINES	0.022517522	31KB
VIA	0.047132996	44KB
TUNELES(5FAMILIAS)	0.059065169	32KB
CON SHADERS		
AGUJAS	0.1496721	32KB
SOLAPAS_ESPADINES	0.125493375	32KB
VIA	0.290442	42KB
TUNELES	0.066277633	32KB
BLOQUES		
CAJAS DE TÚNEL	0.013978124	0.025MB
ESTACIONES	0.010244279	0.012MB
TUNEL_RESOLUCION_3M	0.000922779	0.005MB

Como se puede observar en la siguiente tabla, mientras que un aumento del tamaño del entorno afecta de manera muy limitada en la generación modular (el aumento en el número de familias será el factor determinante), en la construcción clásica por bloques los tiempos de carga y memoria de almacenamiento aumentan de manera proporcional a este incremento de tamaño.

Tabla 4.Memoria de almacenamiento estimados para una línea ferroviaria de 100km y los correspondientes a la Línea 7 de Metro de Madrid (30km)

ELEMENTO	100KM	30KM
SIN SHADERS		
AGUJAS	704KB	704KB
SOLAPAS_ESPADINES	1.4MB	1.4MB
VIA	2MB	2MB
TUNELES(5FAMILIAS)	12MB	1.31MB
CON SHADERS		
AGUJAS	96KB	96KB
SOLAPAS_ESPADINES	256KB	256KB
VIA	128KB	128KB
TUNELES	1.125MB	384KB
BLOQUES		
CAJAS DE TÚNEL	202MB	55MB
ESTACIONES	146MB	44MB
TUNEL_RESOLUCION_3M	439MB	133MB

Sin embargo no se puede pasar por alto que el empleo de shaders implica un consumo de recursos de GPU. Un mayor número de módulos implicará un mayor número de cálculos para la GPU y por tanto un descenso en el rendimiento. La Tabla 5 muestra como la velocidad de renderizado apenas varía cuando se aplican shaders para módulos de 20 metros pero desciende significativamente en el caso de emplear shaders con módulos de 1metro. Queda así nuevamente patente la necesidad de elegir cuidadosamente el tamaño del módulo.

A su vez dicha tabla pone también de manifiesto cómo el empleo de módulos no sólo implica un descenso en la memoria de almacenamiento y los tiempos de carga sino que puede llegar a suponer una mayor velocidad de renderizado que la alcanzada mediante la construcción clásica con bloques de mallas [313]. El aumento en la velocidad de renderizado que supone el empleo de la instanciación está íntimamente ligado a la implementación que el motor gráfico realice de dicha instanciación. Para sacar el máximo partido a la misma DirectX ha desarrollado su API Geometry Instancing [314] mientras que OpenGL ha desarrollado la técnica GLSL Pseudo-instancing [315].

[313] En la prueba reflejada en la Tabla 5 el tamaño de los bloques se elige teniendo en cuenta que un bloque ha de presentar condiciones de iluminación constantes, con el fin de poder estructurarlo adecuadamente en el grafo de la escena y transmitir así al motor gráfico el tipo de iluminación necesaria.

[314] Pharr, M., Randima, F. Chapter 3: Inside geometry instancing.GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation. Addison-Wesley Professional 2005. ISBN: 978-0321335593

[315] Zelsnack, J. 2004. GLSL pseudo-instancing. Technical report., NVIDIA Corporation.

Tabla 5.Resultados obtenidos al generar la vía de la Línea de Metro Ligero de San Chinarro por diversos procedimientos.

TIPO DE CREACIÓN	FrRt (fr/seg)	Nº drawables visualizados [316].	Nº vértices visualizados.
MÓDULOS 20M			
MEDIA	57.4240281	1096	140,104
MIN	24.6362	722	99,151
MAX	101.419	1471	181,056
SHADERS 20M			
MEDIA	56.6459284	708	85,123
MIN	24.2026	32	66,394
MAX	127.775	1384	103,851
MÓDULOS 1M			
MEDIA	48.0596126	2478.5	216,644
MIN	5.24856	1156	199,243
MAX	110.001	3801	234,045
SHADERS 1M			
MEDIA	26.0327378	1982.5	201,682
MIN	1.72306	610	179,480
MAX	106.72	3355	223,884
BLOQUES RESOLUCIÓN 1M			
MEDIA	55.8306461	1065.5	144,029
MIN	23.8267	724	106,977
MAX	102.795	1407	181,080

5.5.3.5 DEFINICIÓN DE GRUPOS

La presente Tesis ha empleado los grupos en diversas situaciones:

- **Jerarquización del entorno para facilitar la localización de elementos**: para agilizar la localización de elementos en el escenario y la edición de los mismos, la Tecnología Modular estructura el entorno atendiendo a su carácter uni o bidimensional:
  - En entornos ferroviarios, donde prevalece el carácter unidimensional, el escenario se divide en tramos de estación y tramos de interestación (Figura 5.51).
  - En entornos urbanos, donde prevalece un carácter bidimensional, el escenario se organiza en una estructura reticular.
- **Agrupación de elementos según su iluminación**: con el fin de facilitar las operaciones de iluminación que el motor gráfico ha de llevar acabo, los objetos del escenario se clasifican en Interiores y Exteriores. Para establecer dicha clasificación, cada familia modular se define como familia de interior, exterior o de ambos tipos 317. En éste último caso será necesario determinar la interioridad de este elemento en función de su PK. En determinados escenarios, bajo condiciones particulares podrán definirse nuevos grupos con características especiales de iluminación.

[316] Un drawable es una geometría que requiere para su representación de una llamada a una función Draw* (dependiendo del API elegida esta función se llamará de una forma u otra). Cada llamada implica un consumo adicional de recursos, por este motivo se intenta agrupar las geometrías en el mínimo número de drawables posibles.

[317] Ver archivo de configuración cfg_TipoSubfamilias.txt en el Anexo 3.

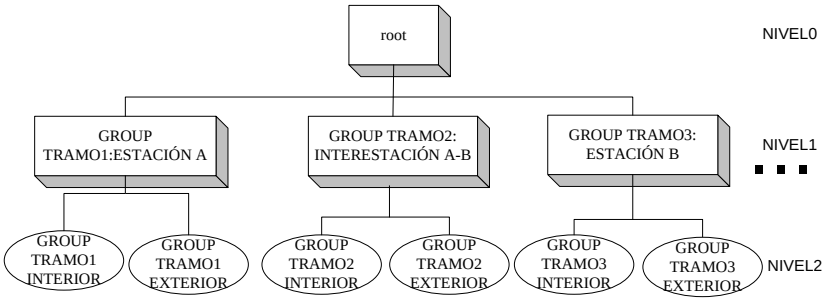


Figura 5.51.Agrupación de elementos según su posicionamiento respecto de estaciones e iluminación.

- **Agrupación de elementos con el mismo comportamiento:** para poder representar diversos comportamientos los objetos del escenario se agrupan en switches. En este esta Tesis los nodos switch se han empleado en dos situaciones:
  - Simulación del comportamiento de agujas: las agujas podrán tener un estado directo y un estado desviado. Como se comentó en el Capítulo 3 [318], la representación virtual de una aguja se realiza a través de tres familias de módulos: raíles, solapas y espadines. Las solapas y espadines serán los elementos móviles y serán incluidos en un switch.
  - Simulación del comportamiento de semáforos: los distintos estados que puede representar un semáforo se codificarán en distintos estados del switch. Buscando la máxima flexibilidad constructiva y la mínima carga de modelado, los semáforos se generan combinando diversos tipos de familias de módulos: postes, carcasas y luces. En el Anexo 7 se detallan cada una de ellas así como la metodología y convenios adoptados para su uso y representación.

5.5.3.6 NÚMERO DE LODS.

Con el fin de descargar al máximo la CPU, algo de vital importancia en una simulación de conducción, la Tecnología Modular optó por un sistema de niveles de detalle discretos. Aprovechando la condición de todo simulador de conducción terrestre guiada, de seguir una serie de trayectorias predefinidas, la Tecnología Modular añade a este sistema de LODs discretos, la condición de ser variante con el punto de vista. Para conseguir esto último se realiza una discretización transversal del entorno, de manera que las zonas más próximas a las trayectorias se generen con el máximo nivel de detalle y éste vaya disminuyendo a medida que la distancia aumenta. Las distancias de corte transversal se determinan experimentalmente, siendo la amplitud del campo de vista el factor determinante. Este es el motivo por el que se ha definido el sistema de LOD propuesto por esta Tesis como pseudo-variante con el punto de vista, por encontrarse en un término medio entre el sistema de LOD discreto en que no existe dicha variación y el sistema de LOD continuo en el que esta variación se produce a nivel de vértice.

Este sistema de LODs se materializa en el grafo de la escena mostrado en la Figura 5.52. Este grafo consta de dos niveles jerárquicos de tipo LOD anidados. El primer nivel (Nivel 3) está constituido por nodos de tipo "se ve o no se ve" y permitirá descartar de cálculos posteriores todas aquéllas zonas del escenario que exceden la distancia de LOD elegida. El segundo nivel (Nivel 4) está formado por los nodos terminales L1,L2, que contendrán las geometrías de los módulos con diferente número de polígonos.

[318] Ver Capítulo 3 apartado 3.6.1.

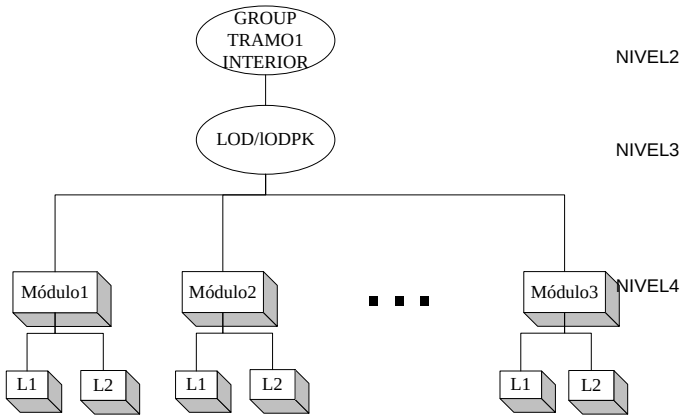


Figura 5.52.Ejemplo uso de niveles de detalle.

Si se tienen los siguientes valores:

- Plano de yon situado a *y* metros.
- Longitud de módulo : *Lb* metros.
- Un LOD (LOD padre) por cada N_{LOD} módulos.
- Longitud total del entorno: *L metros*

La distancia de LOD mínima que permite en todo momento que el entorno desaparezca a consecuencia del plano de yon será:

$$d\_lod = y + \frac{(Lb \cdot \mathrm{N}_{\mathrm{LOD}})}{2}$$

Con lo que el número de operaciones vendrá dado por:

$$Num\_operaciones = \frac{L}{(Lb \cdot \mathrm{N}_{\mathrm{LOD}})} + \mathrm{int}\left(\frac{y}{Lb \cdot \mathrm{N}_{\mathrm{LOD}}} + 1\right) \cdot \mathrm{N}_{\mathrm{LOD}}$$

El primer término de la expresión anterior representa el número de LODs padre presentes en el entorno. El segundo se corresponde con al número de operaciones asociadas a cada uno de los hijos de los LODs padre cuya distancia al punto de vista es menor que *d_LOD*.

Por tanto, el número de módulos óptimo se obtendrá derivando la expresión anterior e igualándola a 0.

$$\frac{-L}{\mathrm{N}_{\mathrm{LOD}} \cdot \mathrm{N}_{\mathrm{LOD}} \cdot Lb} + 1 = 0$$

De forma que, el número de elementos que cuelgan de un LOD padre vendrá dado por la siguiente expresión:

$$N_{LOD} = \sqrt{\frac{L}{Lb}}$$

el cuál debe ser un parámetro de diseño a la hora de crear la jerarquía.

5.6 EJEMPLOS DE APLICACIÓN.

A continuación se presentan una serie de ejemplos en los que la Tecnología Modular ha sido empleada con gran éxito. Dependiendo de las características del proyecto y las exigencias de similitud con la realidad, los entornos virtuales generados para los simuladores de conducción del CITEF incorporan en mayor o menor medida la generación modular.

La Tecnología Modular facilita la creación de los escenarios de manera escalonada, añadiéndose precisión a los mismos de manera paulatina en función de los requerimientos del cliente. De esta manera éste puede disponer desde el instante inicial de una versión piloto del entorno a simular. Las etapas a seguir por la Tecnología Modular en la creación de un entorno ferroviario son ser las siguientes:

- **Etapa 1**: generación de dos líneas paralelas consolidadas. Se introducen estaciones genéricas y una familia de terreno o túnel genérico dependiendo de si se trata de un entorno predominantemente exterior o interior.
- **Etapa 2**: introducción de intervalos de paralelismo, independencia y conexión.

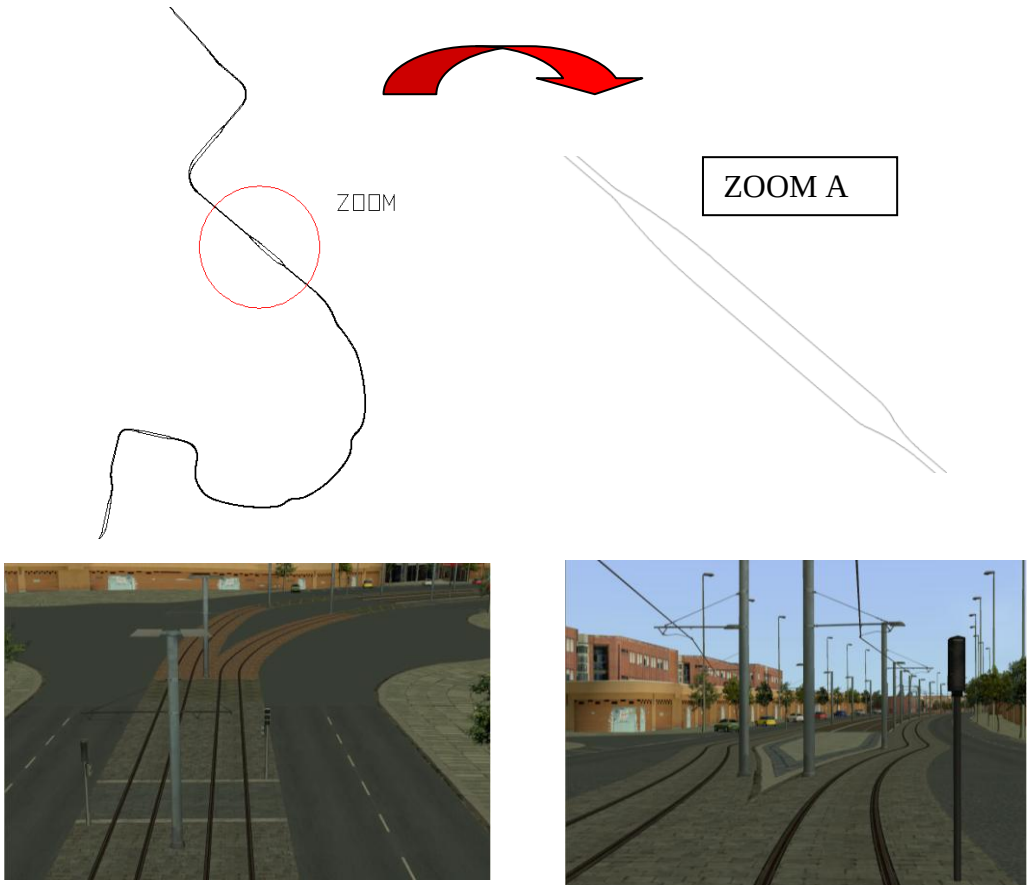


Figura 5.53.Trazado ferroviario: visión esquemática y 3D de intervalos de paralelismo, independencia y conexión.

- **Etapa 3**: introducción de nodos circulatorios ferroviarios: líneas secundarias y cambios de vía.

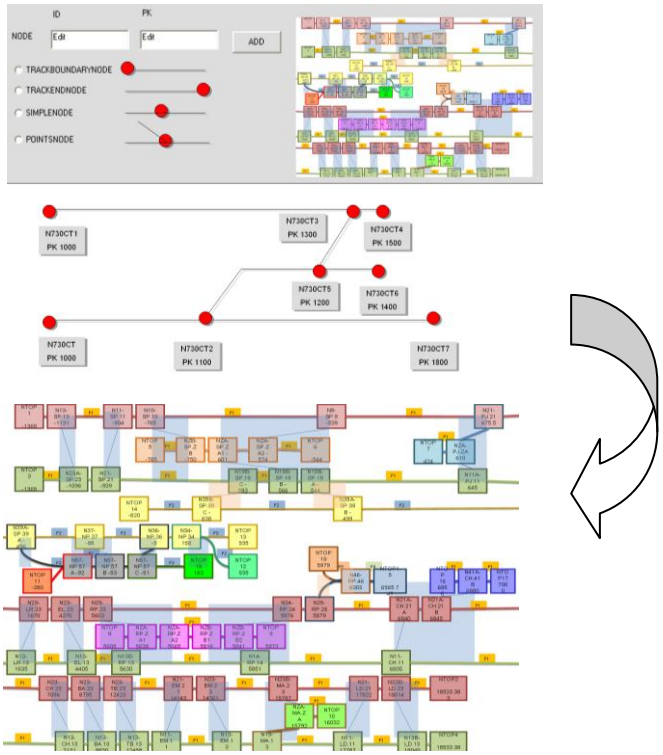


Figura 5.54.Generación de la información de nodos de las líneas de Metro Ligero de Madrid.

- **Etapa 4**: introducción de la red circulatoria urbana. El enorme volumen de información a manejar en el caso de simulaciones ferroviarias que tienen lugar en entornos urbanos, como es el caso de las líneas de Metro Ligero de Madrid, obliga a intensas tareas de consolidación que garanticen las relaciones de paralelismo, cortes y cruces entre ambos entornos, ya que la información del entorno ferroviario y urbano suele proceder de fuentes distintas y suelen ser numerosas las incongruencias entre ellas.

  De esta manera, una vez definido el trazado ferroviario se posicionan los nodos de la red urbana. Los nodos que se encuentran sobre las arterias principales de la ciudad, paralelas a la red ferrovaria, se calculan automáticamente teniendo en cuenta su PK y su distancia a la misma. Los restantes nodos y calles se introducen con la ayuda de las herramientas desarrolladas por esta Tesis (Figura 5.55), tomando como base la red de trayectorias construida hasta el momento y los planos topográficos disponibles.

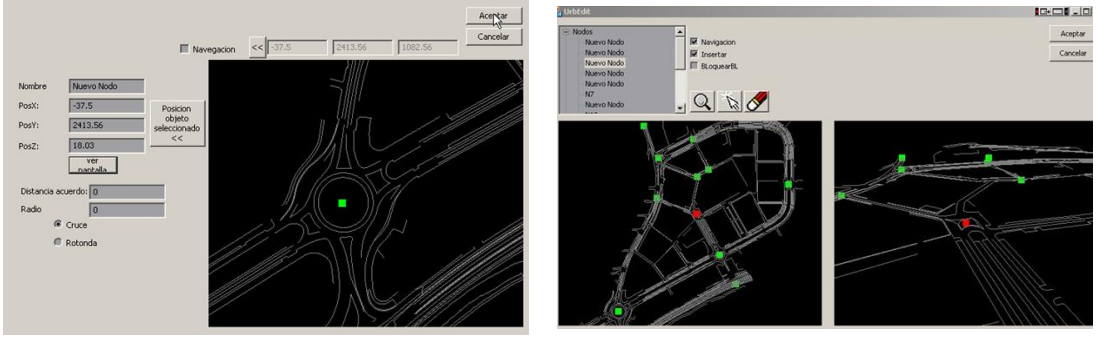


Figura 5.55.Interfaz de entrada de nodos de Urbedit.

A continuación la Tecnología Modular genera información circulatoria (trayectorias y regulación semafórica y vial) coherente para ambos entornos. La siguiente figura muestra imágenes del trazado ferroviario y la red urbana generada para Metro Ligero de Madrid.

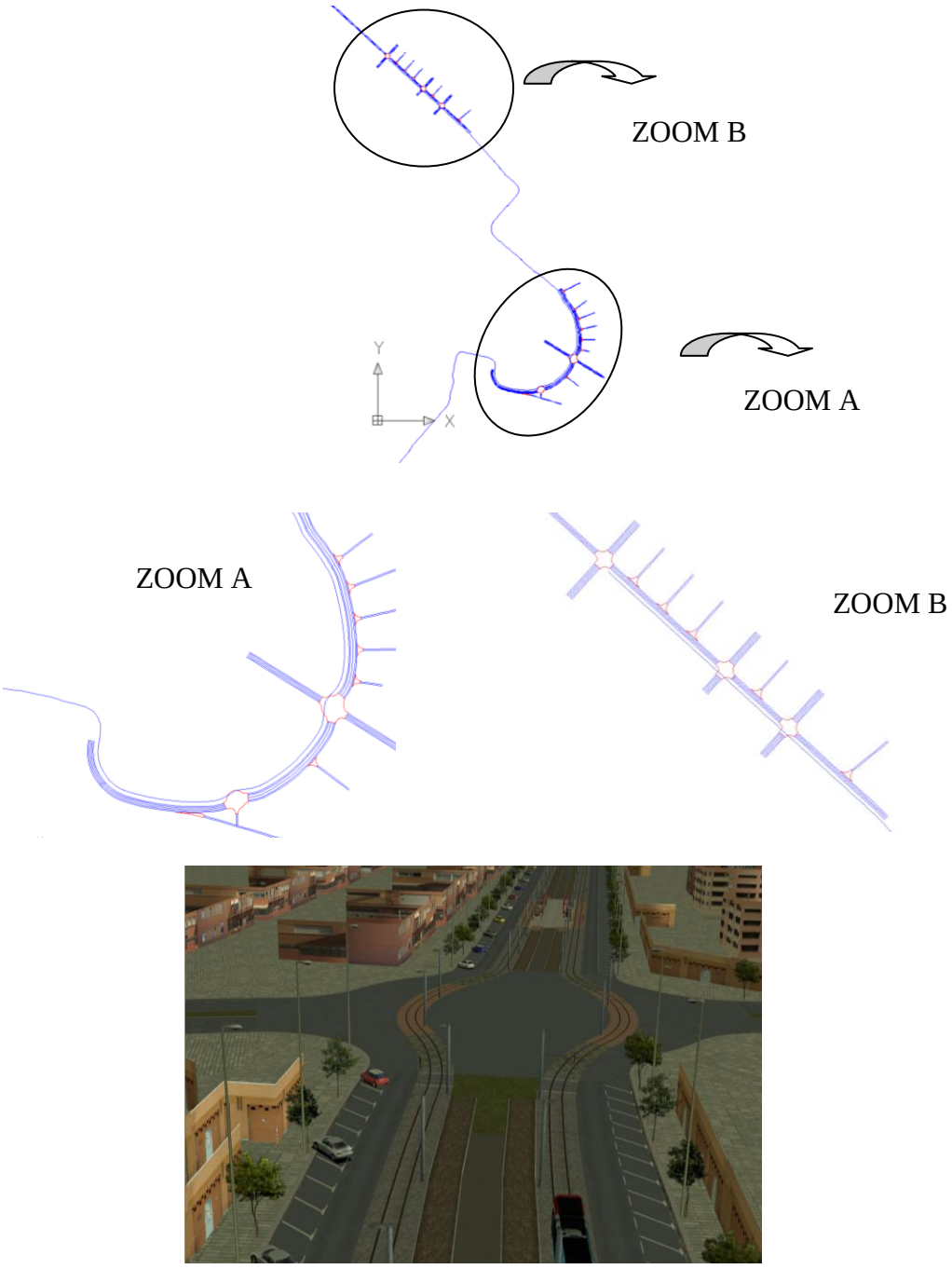


Figura 5.56.Trazado generado para la línea ML1 de Metro Ligero de Madrid.

Para cada intersección se genera también la información de todas las posibles rutas que un vehículo puede seguir en la misma (Figura 5.57). Dicha información se envía al subsistema de tráfico urbano siguiendo los formatos acordados [319] .

[319] Anexo 2.

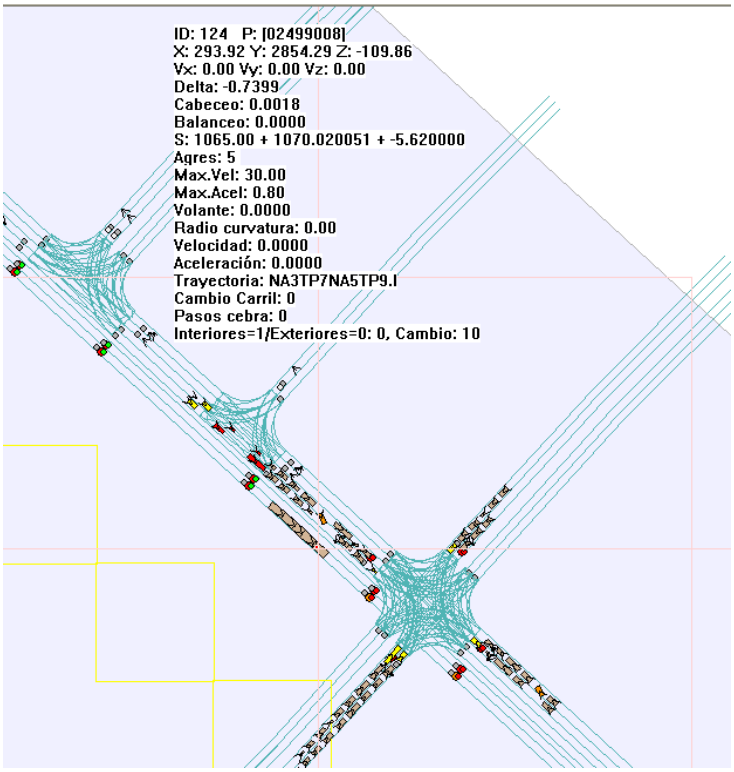


Figura 5.57.Ejemplo de rutas vehiculares generadas para Metro Ligero de Madrid: aplicación MicroTraff desarrollada por el Grupo de Ingeniería Gráfica y Simulación para el control del tráfico vehicular y visual.

- **Etapa 5**: introducción de información de entornos. La definición del entorno consiste en la asignación de familias modulares a todas aquéllas zonas que verifican los requisitos exigidos por la Tecnología Modular: existencia de una dirección privilegiada entorno a la cual se extiende el entorno y posibilidad de sintetización en base a una serie de patrones repetibles. Las zonas en las que no sea posible la generación modular se generarán con mallas (Figura 5.58). Tanto la asignación de familias como la selección de los contornos de las mallas a generar se realizará con la ayuda de las herramientas desarrolladas por esta Tesis [320].

Figura 5.58.Ejemplo de combinación de módulos con mallas para la línea ML1 de Metro Ligero de Madrid.

[320] Anexo 6.

Las mallas generadas pueden ser editadas a través de la interfaz de edición. Esta edición incluye la recolocación de elementos anexos, la importación/exportación de elementos a 3D Studio y la modificación de la malla en sí, cambiando ya sea el contorno asignado o el ángulo de suavizado del mismo (Figura 5.59).

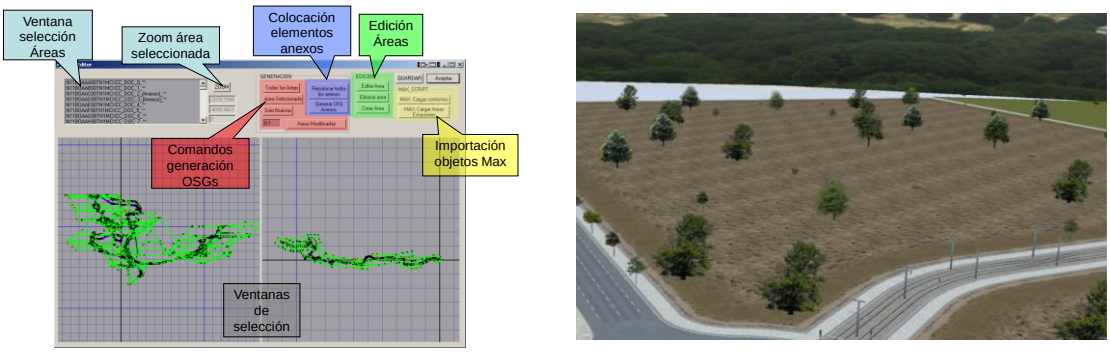


Figura 5.59.Interfaz de edición de mallas.

- **Etapa 6**: finalmente se definen los parámetros de optimización gráfica:

  - **Longitud básica**: Un análisis del comportamiento de la velocidad de renderizado en función del número de instancias servirá como punto de partida para determinar la longitud modular básica y parámetros del grafo de la escena, como el número y tamaño de los grupos y niveles de detalle necesarios. Se parte para ello del entorno piloto generado en la Etapa 1 y se obtiene el gráfico mostrado en la Figura 5.60. La similitud de características encontradas entre los diversos entornos generados para los simuladores de conducción ferroviaria del CITEF han llevado a establecer como longitud básica estándar los 20 metros. En algunos casos específicos, como en el simulador desarrollado para metro de Santiago de Chile, la existencia de características repetitivas en los túneles cada 25 metros motivó la elección de esta magnitud como longitud básica.

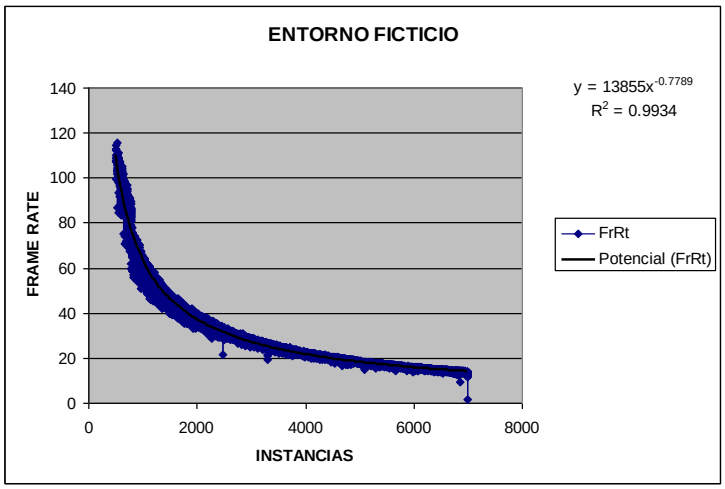


Figura 5.60.Variación de la velocidad de renderizado con el número de instancias en la creación de un entorno ferroviario.

- **Niveles de detalle**: En aquéllos entornos donde se combinan entornos ferroviarios con entornos urbanos como es el caso de las líneas de Metro Ligero de Madrid, se crearán dos archivos de texto plano distintos para guardar por un lado el grafo de la escena del entorno ferroviario y por otro el grafo de la escena del entorno urbano.

  En los entornos ferroviarios se emplearán dos tipos de estructuras LOD diferentes en función del tramo de entorno ferroviario a representar:

  - En el caso de coexistencia de varias líneas ferroviarias donde no exista una línea directriz dominante que defina un único PK, se empleará un LOD que tomará como centro de sus cálculos el centro geométrico del área abarcada por dicho LOD (Figura 5.61).
  - En el caso de zonas donde el entorno posee un carácter predominantemente unidimensional se empleará el llamado LODPK, el cuál emplea como centro geométrico de sus cálculos el punto medio de la línea directriz que marca la unidimensionalidad en el intervalo de pks, lo que simplifica las operaciones.

  En el caso de los entornos urbanos el entorno se estructura en una retícula.

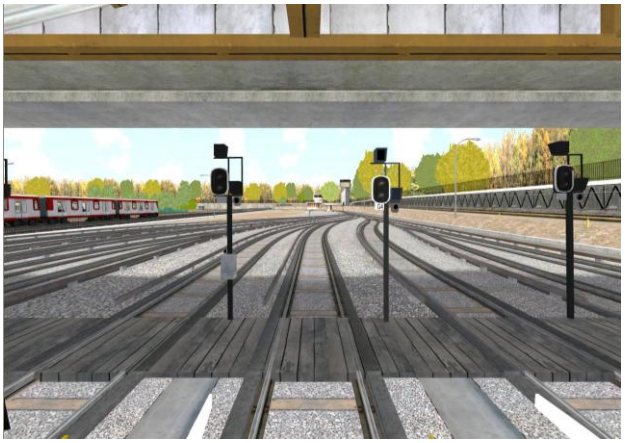


Figura 5.61.Depósito de Metro de Santiago de Chile: ejemplo de zona en la que no es posible el empleo de LODPK.

- **Shaders**: Dependiendo del tipo de entorno a representar los shaders empleados para deformar el módulo en tiempo real podrán ser combinados con otro tipo de shaders, como los de iluminación. La Figura 5.62 muestra el resultado de aplicar shaders al simulador desarrollado para metro de Santiago de Chile.

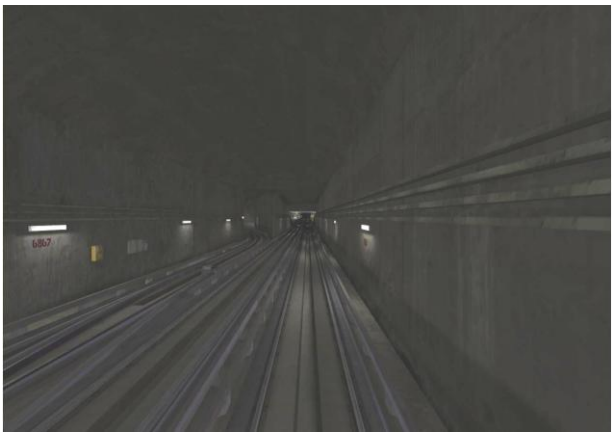


Figura 5.62.Entorno generado mediante el empleo de shaders para Metro Santiago de Chile.

A través de las herramientas vistas es posible también la generación de entornos ficticios. Las familias a emplear podrán seleccionarse de entre las base de datos modulares ya creadas o generarse otras nuevas a través de la herramienta GENFAM [321], ya sea partiendo desde cero o por combinación y /o edición de las familias ya existentes. A través de GENENT [322] podrá introducirse el posicionamiento de estas familias en el escenario. A través de la herramienta URBEDIT se facilita la entrada de información relacionada con un entorno urbano.

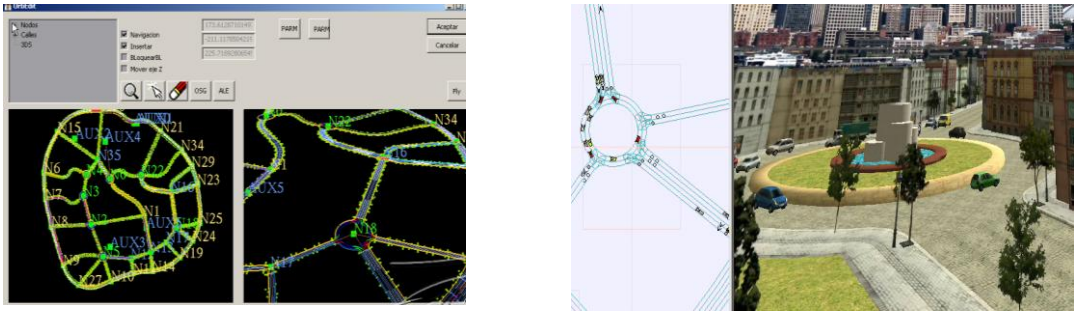


Figura 5.63.URBEDIT: herramienta para la entrada de datos de un entorno urbano.

La Figura 5.64 muestra distintos ejemplos de ciudades ficticias generadas con la Tecnología Modular.

Figura 5.64.Ejemplos de ciudades ficticias desarrolladas por la Tecnología Modular.

[321] Anexo 3.

[322] Anexo 6.

5.7 PROYECTOS EN LOS QUE HA PARTICIPADO LA TECNOLOGÍA MODULAR.

Los inicios de la Tecnología Modular tuvieron lugar alrededor del año 2000. El proyecto "GifTren, Diseño de Instalaciones y la Simulación de la Explotación de Líneas Ferroviarias de Alta Velocidad", bajo acuerdo con el Gestor de Infraestructuras Ferroviarias (GIF) (Ministerio de Fomento) fue su primera aplicación. El objetivo de este proyecto era la generación de una herramienta informática que permitiese el diseño de las infraestructuras de una línea ferroviaria y el análisis de la explotación de la misma. El programa tenía en cuenta factores como son:

- Las características del material rodante: dimensiones, curvas de tracción, el número de pasajeros, los tipos de freno, el peso freno, en general todas las características que definen el comportamiento del material en la explotación de una línea.
- Las características de la circulación: el horario, el itinerario, etc.
- Señalización: ERTMS nivel 2 y 3.
- Las características de la infraestructura de la línea: estaciones, trayectos, aparatos de vía y bypass.

Los resultados de la aplicación GifTren se materializaron en una simulación virtual en la que se representaba una línea ferroviaria y se permitía el movimiento de trenes, cuyo comportamiento se veía plasmado en diversos tipos de gráficas.

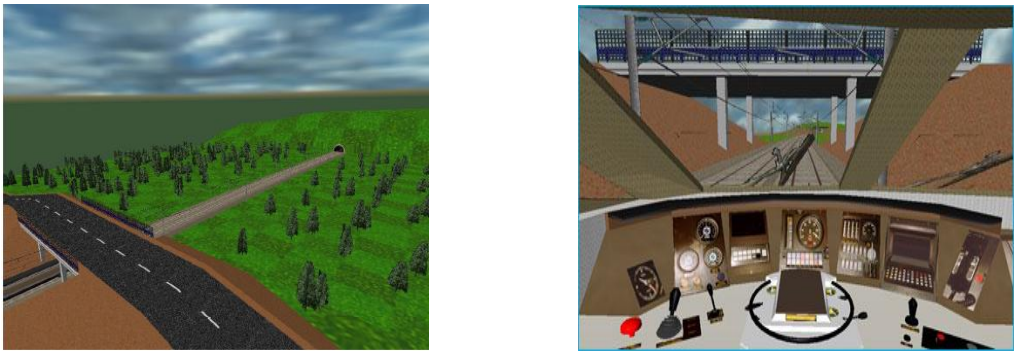


Figura 5.65.Entorno generado con la Tecnología Modular en GifTren.

Los entornos fueron generados modularmente y la definición de los mismos se realizó a través de una aplicación diseñada para ello (Figura 5.66).

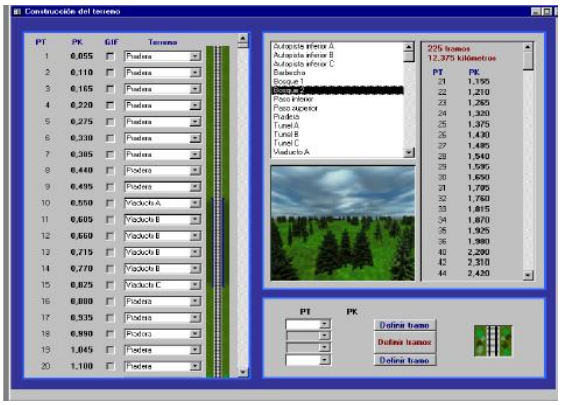


Figura 5.66. Aplicación para la definición de entornos en GifTren .

En aquella época la creación de los módulos era completamente manual. El trazado solo permitía la inclusión de radios y rectas. Los algoritmos de posicionamiento modular estaban en sus inicios:se comienzan las primeras pruebas del Algoritmo Inscrito.

En el año 2001 la Tecnología Modular participó en el Programa PROFIT, con el proyecto "Desarrollo de un Sistema Distribuido para la Emulación de Conducción de Trenes de Alta Velocidad Bajo Sistema ERTMS y Entorno de Realidad Virtual".

Los objetivos de este proyecto eran el desarrollo e integración de un conjunto de tres cabinas simuladoras de conducción de trenes de alta velocidad bajo el sistema ERTMS. Se buscaba experimentar la interoperabilidad del sistema ERTMS (línea Madrid-Sevilla con línea Madrid-Barcelona) y servir como medio de formación y entrenamiento de maquinistas en ERTMS (Figura 5.67). Además se pretendía validar reglas y procedimientos en dicho estándar (proyecto HEROE, Harmonization of European Rail Rules for Operanting ERTMS)

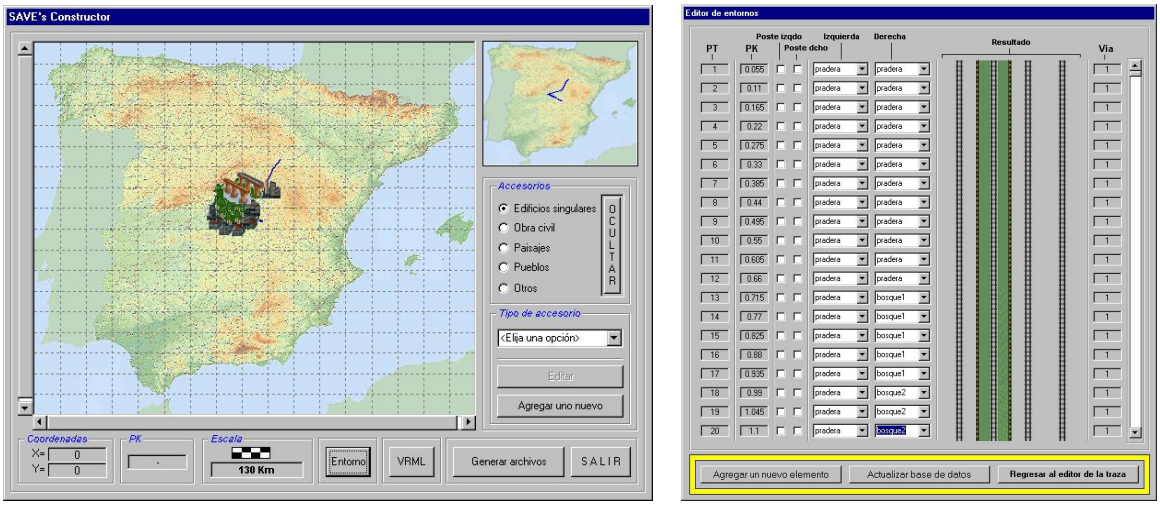


Figura 5.67.SAVE: Aplicación para la gestión de entornos virtuales para el Simulador del AVE.

En esos tiempos la Tecnología Modular ofrecía poca flexibilidad. El módulo tenía una longitud constante de valor 55 metros (Figura 5.68) y no existía división transversal del entorno, sino que un mismo módulo abarcaba todos los niveles de proximidad.

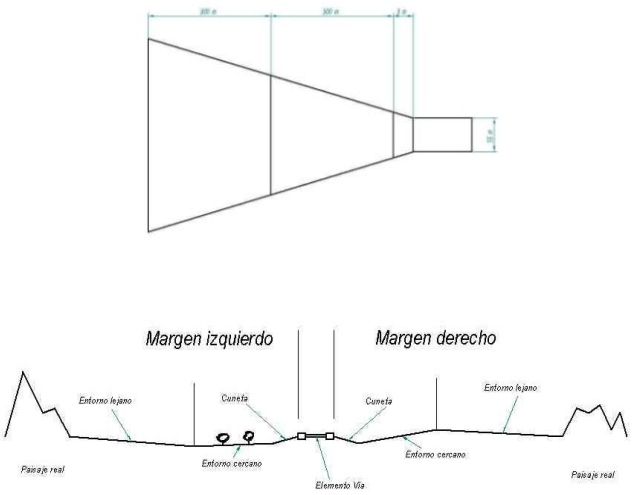


Figura 5.68.Módulo de 55metros con todos los niveles de proximidad incorporados; 3 niveles a 3, 300 y 600 metros de distancia.

Los resultados de la metodología empleada en ese momento se describieron en los artículos presentados en Ingegraf 2001 [323] [324] (Figura 5.69).

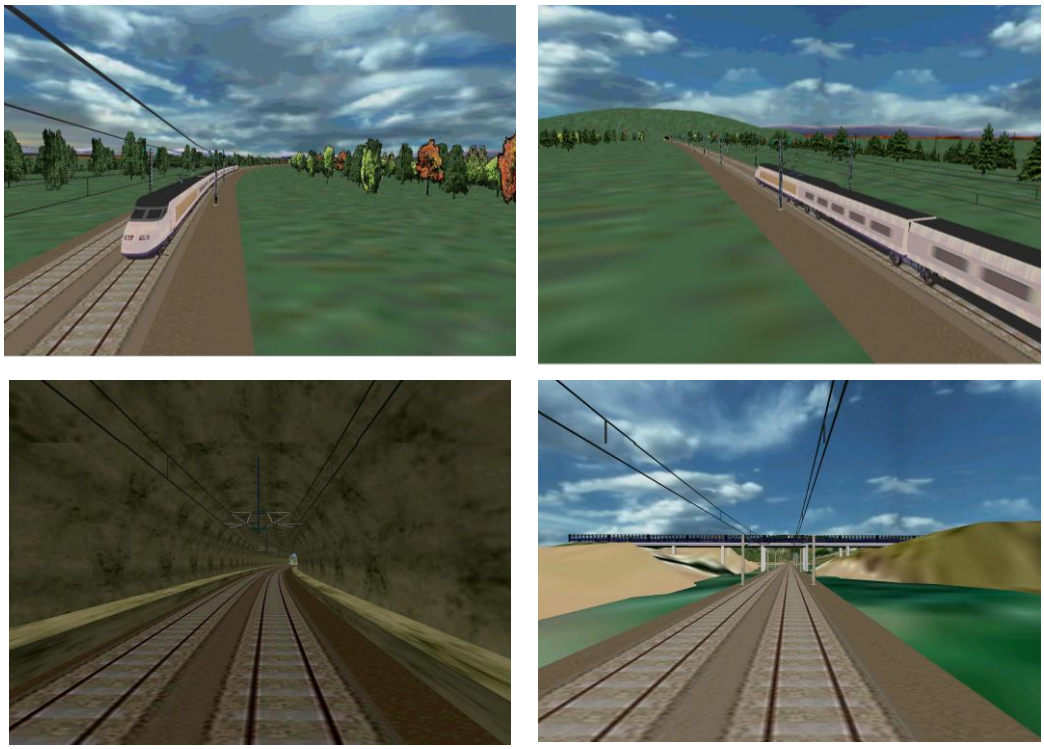


Figura 5.69. Escenarios desarrollados por la Tecnología Modular para el AVE.

Entre los años 2001 y 2003 la Tecnología Modular participa en el proyecto para Metro de Madrid "Simulador de Conducción y Averías de METRO de Madrid SA: Escuela de Simulación de Trenes 7000 y 8000", que el CITEF desarrolla en colaboración con Indra. Los objetivos de este proyecto eran la formación de conductores en el manejo de las nuevas Unidades Tren, la resolución de averías en las mismas en línea, la conducción con los nuevos sistemas de protección en las líneas 8, 9, 10 y MetroSur y la preparación frente a incidencias de circulación.

Para conseguir estos objetivos se simulan el comportamiento y averías de los nuevos materiales móviles y los sistemas neumáticos, eléctricos, electrónicos y lógicos de las series 7000 y 8000 y los sistemas de control y protección de trenes así como el programa de explotación de los enclavamientos de las Líneas 8, 9, 10 y Metro Sur.

La Tecnología Modular participará en la generación de los escenarios de estas líneas de Metro: alrededor de 90 estaciones y 150 km de red.

La necesidad de representar trazados topográficos que los conductores pudiesen reconocer obliga al desarrollo del Algoritmo de Consolidación, que se utiliza para garantizar la congruencia trazado real- trazado simulado de todas estas líneas. El Algoritmo Inscrito queda completamente

[323] Tovar, C; Félez, J; Cabanellas, J.Mª ; Romero, G. Generación automática de entornos para aplicaciones realidad virtual. XIII Congreso Internacional de Ingeniería Gráfica. Badajoz. INGEGRAF 2001.

[324] Cabanellas, J. Mª, Félez, J., Martinez, Mª Luisa, Carretero , A., Tovar,C. Visualizadores de realidad virtual para un simulador de vehículos distribuido. XIII Congreso Internacional de Ingeniería Gráfica. INGEGRAF 2001.

definido y asentando con la generación de estas líneas de metro. Los desarrollos llevados a cabo en estos años de proyecto se reflejan en el artículo publicado en Ingegraf 2003 [325] (Figura 5.70).

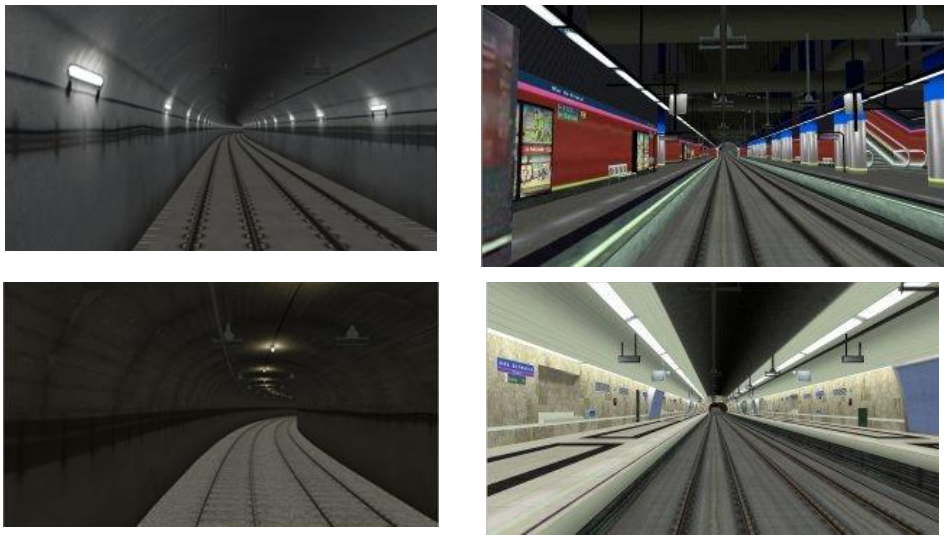


Figura 5.70. Ejemplos de las líneas de Metro generadas hasta el año2003.

Entre los años 2002-2006 CITEF desarrolla el proyecto "ERTMS Technology Centre, Simulador ERTMS para Prueba de Equipos", bajo encargo de la empresa Invensys (Reino Unido), a través de sus filiales Westinghouse Rail Systems Limited (Reino Unido) y Dimetronic Signals S.A. (España) . Los objetivos de este proyecto eran el desarrollo de un sistema de simulación completo del sistema ERTMS según la versión 2.2.2 de la especificación, en sus niveles 0, 1 y 2, y de una herramienta para verificación, puesta en marcha y mantenimiento de equipos de señalización interoperables ERTMS. Dentro de la estructura del simulador del sistema ERTMS se proponía sustituir elementos simulados, por elementos reales. De esta forma sería posible probar los equipos desarrollados dentro de un entorno sin riesgos y con mucho menor coste.

A su vez se buscaba configurar distintas líneas y escenarios para definir a nivel funcional los elementos e información a instalar. En la tarea de configuración de escenarios es dónde interviene la Tecnología Modular, creando un editor de entornos (Figura 5.71) que permitía la configuración y representación de entornos ficticios mediante el empleo de módulos. De esta forma se ofreció un sistema visual de realidad virtual, que permitía reconfigurar la línea a representar y representar en tiempo real el estado de la línea y los elementos presentes (Figura 5.72).

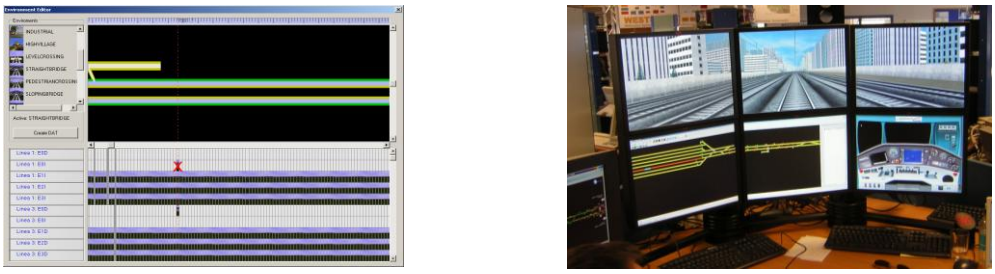


Figura 5.71.Editor de entornos desarrollado por la Tecnología Modular para Invensys.

[325] Tovar, C., Cabanellas, J.Mª, Félez, J. Procedimientos geométricos para el ajuste de trayectorias ferroviarias en simuladores 3D de diseño modular. XV Congreso Internacional de ingeniería Gráfica. INGEGRAF 2003.

La generación de módulos seguía sin estar automatizada lo que ralentizaba y restaba flexibilidad a la generación de entornos. Se define una nomenclatura de módulo que permita facilitar la búsqueda de compatibilidades entre familias a la hora de crear entornos ficticios. Sin embargo, la gran cantidad de trabajo que es necesario hacer manualmente es fuente de errores que dificultan el proceso constructivo.

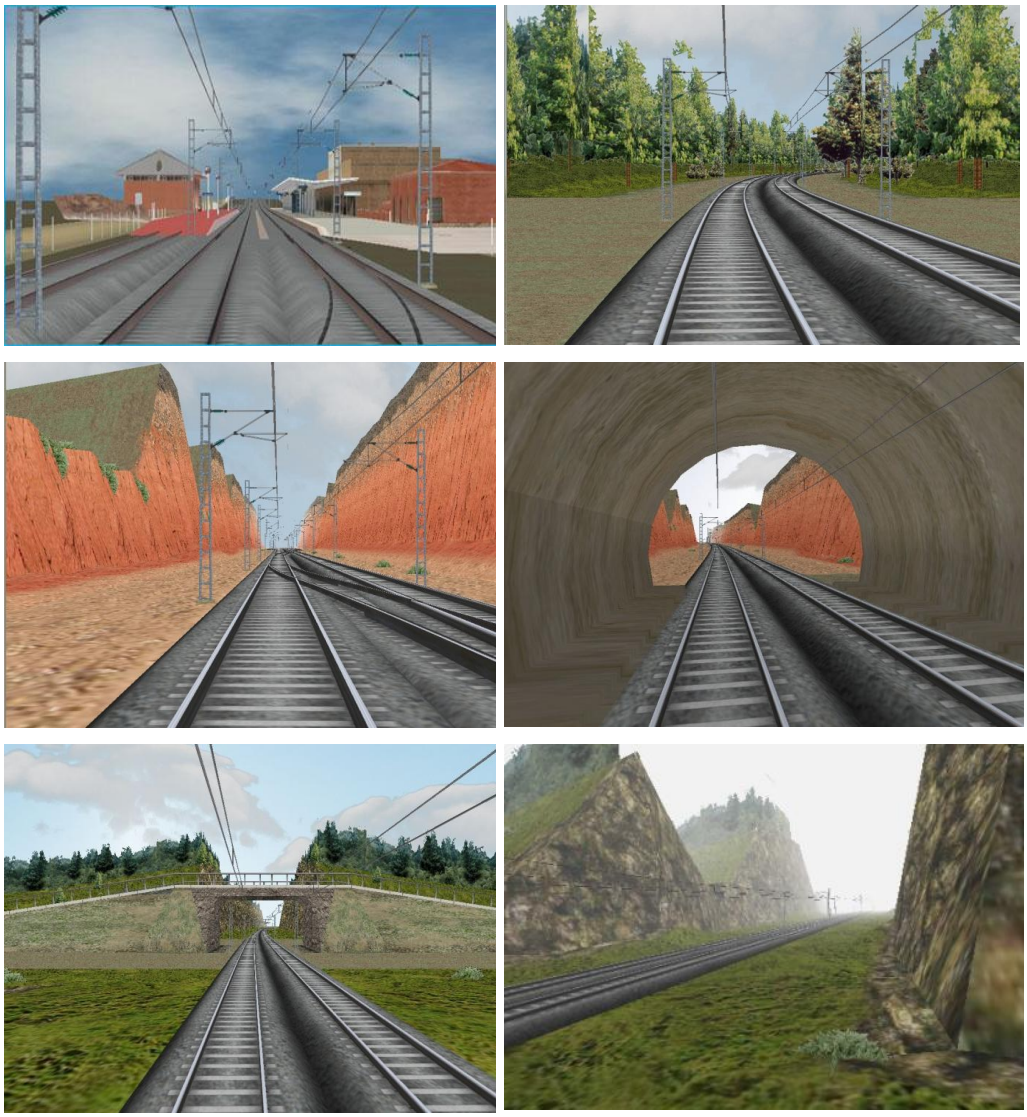


Figura 5.72. Ejemplos de entornos generados por la Tecnología Modular para Invensys

Entre los años 2004 y 2008, se realiza otro proyecto para Metro de Madrid "Simulador de Conducción y Averías de METRO de Madrid SA, Plan de Ampliación 2003 – 2007, Serie 9000". El objetivo de este proyecto era la formación de conductores en el manejo de las nuevas Unidades Tren serie 9000, la resolución de averías en las mismas en línea, la conducción con los nuevos sistemas de protección en la línea 7 y la preparación frente a incidencias de circulación.

Para conseguir estos objetivos se simulan el comportamiento y averías de los materiales móviles y los sistemas neumáticos, eléctricos, electrónicos y lógicos de la serie 9000 y los sistemas de control y protección de trenes así como el programa de explotación de los enclavamientos de la Línea 7.

La Tecnología Modular participará en la generación de los escenarios de esta línea de Metro: alrededor de 25 estaciones y 25 km de red (Figura 5.73).

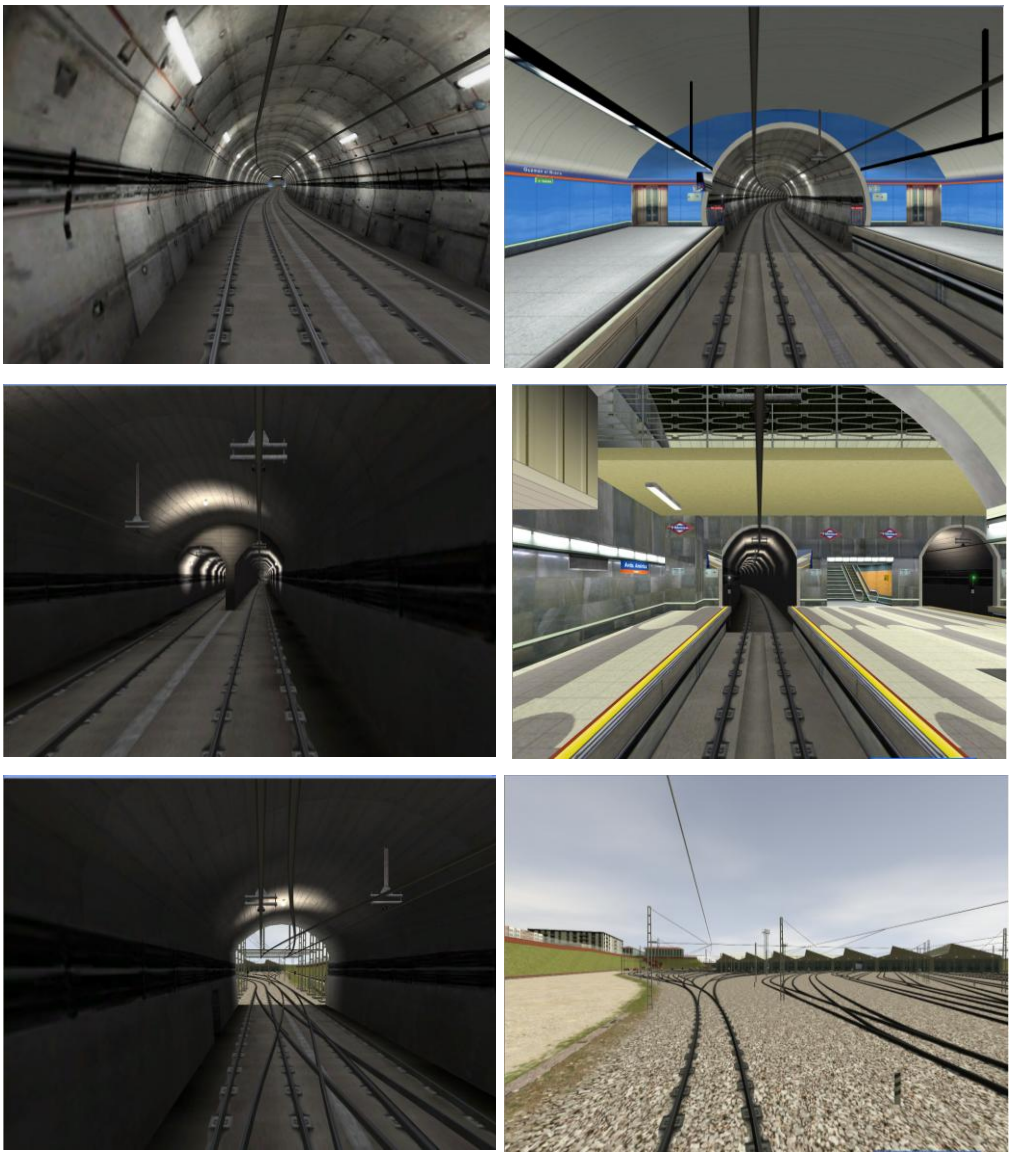


Figura 5.73.Escenarios de la Línea 7 de Metro de Madrid creados por la Tecnología Modular.

Se trata de una línea compleja en la gestión de la información debido a la combinación de pks crecientes y decrecientes y al empleo de túneles de enlace para unir distintos tramos de la misma. Estas zonas de unión suelen ir acompañadas de zonas de playas de vías que exigen introducir modificaciones en la manera unidimensional en la que hasta ahora se trataban los escenarios ferroviarios. Para ello se definen dos tipos de LOD: el LOD PK, que es el que se venía usando normalmente en trazados ferroviarios y que dependía únicamente del pk recorrido y el LOD *centro geométrico*, que permite visualizar sin saltos bruscos zonas de coexistencia de varias lineas principales y por tanto de varios pks.

El desarrollo del Módulo de Ajuste Geométrico con sus algoritmos de suavizado permite definir mediante biarcos toda la línea, empleándose así como algoritmo de posicionamiento el Algoritmo Circunscrito.

La intervención de la aplicación GENFAM para la generación automática de módulos permite crear únicamente aquéllos módulos que el escenario requiera, en lugar de toda la familia base, disminuyendo el espacio de almacenamiento requerido para la base de datos visual final.

Solapándose con este proyecto, entre los años 2005 y 2008, la Tecnología Modular participa en otro proyecto con Metro de Madrid, el "Simulador de Conducción y Averías de METRO de Madrid SA, Plan de Ampliación 2003 – 2007, serie 3000". Este proyecto tiene como objetivos la formación de conductores en el manejo de las nuevas Unidades Tren serie 3000, la resolución de averías en las mismas en línea, la conducción con los nuevos sistemas de protección en la línea 3 y la preparación frente a incidencias de circulación

Para conseguir estos objetivos se simulan el comportamiento y averías de los materiales móviles y los sistemas neumáticos, eléctricos, electrónicos y lógicos de la serie 3000 y los sistemas de control y protección de trenes así como el programa de explotación de los enclavamientos de la Línea 3.

La Tecnología Modular participará en la generación de los escenarios de esta línea de Metro: alrededor de 15 estaciones y 20 km de red (Figura 5.74).

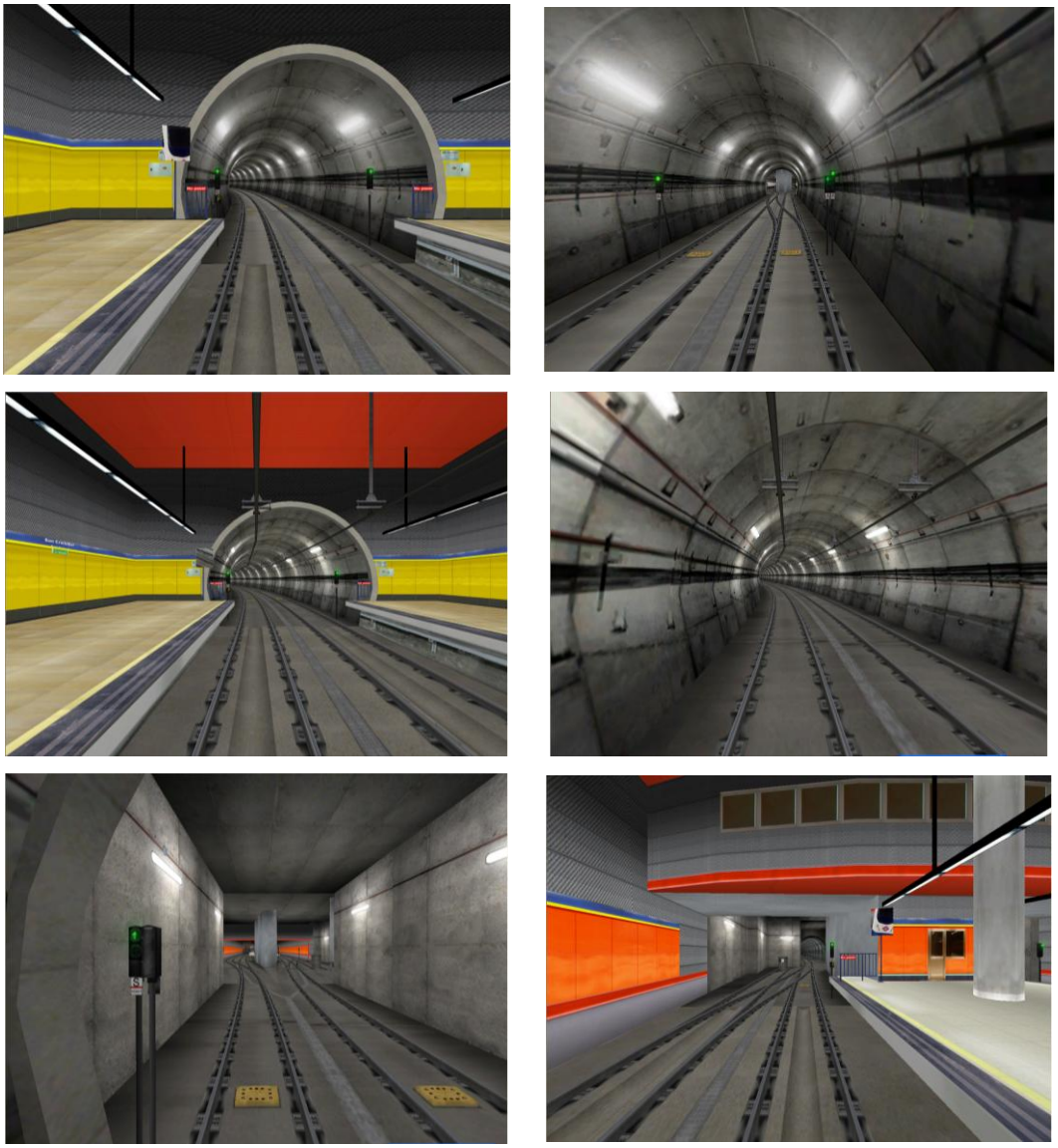


Figura 5.74.Escenarios de la Línea 3 de Metro de Madrid creados por la Tecnología Modular.

Entre los años 2006 y 2008, la Tecnología Modular colabora en el proyecto para Metro de Madrid, "Simulador de Conducción y Averías de Metro Ligero, MINTRA Plan de Ampliación 2003 – 2007, Serie Citadis". El objetivo de este proyecto es la formación de conductores en el manejo de las nuevas Unidades Tren serie Metro Ligero – Citadis y de otros agentes del Metro Ligero: operadores del Centro de Control y SAE, Controladores de Línea, etc, la resolución de averías en las mismas en línea, la conducción con los nuevos sistemas las líneas ML1, ML2 y ML3, la preparación frente a incidencias de circulación e interacciones con el tráfico rodado y los peatones.

Para conseguir estos objetivos se simulan el comportamiento y averías de los materiales móviles y los sistemas neumáticos, eléctricos, electrónicos y lógicos de la serie Citadis y los sistemas de control y protección de trenes así como el programa de explotación de los enclavamientos de las líneas ML1, ML2 y ML 3.

La Tecnología Modular participará en la generación de los escenarios de estas líneas de Metro Ligero: alrededor de 45 estaciones y 45 km de red, con la novedad de incorporar entornos urbanos.

La herramienta URBEDIT agiliza enormemente la entrada de datos de ciudad. Se aplica también por primera vez la metodología de generación automática de mallas en las zonas del entorno que no cumplen los requisitos de modularidad.

Los resultados de la metodología empleada en ese momento se describieron en los artículos presentados en Ingegraf 2006 [326] e Ingegraf 2008 [327].

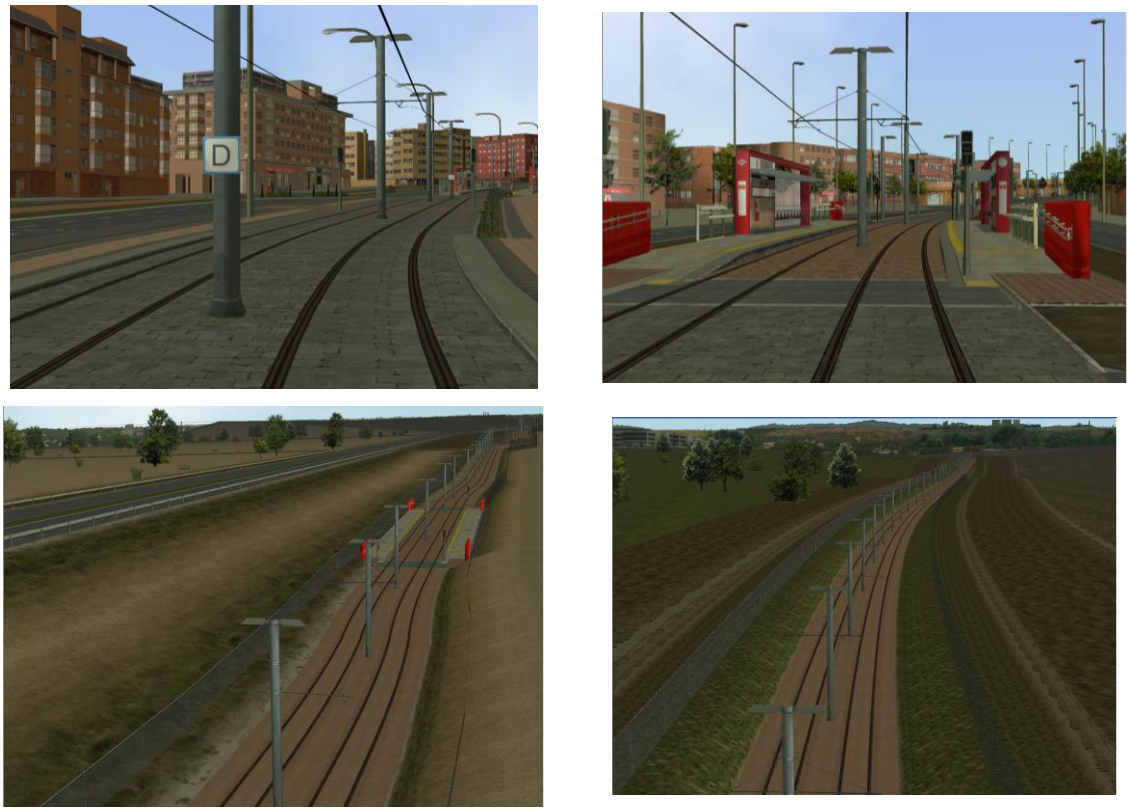


Figura 5.75.Escenarios de la línea ML3 de Metro Ligero de Madrid.

[326] Roel, M., Tovar, C., Cabanellas, J.Mª. Metodología de generación automática de trayectorias de grandes entornos virtuales gestionando errores e incongruencias de los datos de la realidad. XVIII Congreso Internacional de ingeniería Gráfica. INGEGRAF 2006.

[327] Tovar, C., Jimena de Dios, G., Cabanellas, J.Mª, Félez, J. Tecnología modular orientada a shaders en la generación de grandes entornos para simuladores. XX Congreso Internacional de ingeniería Gráfica. INGEGRAF 2008.

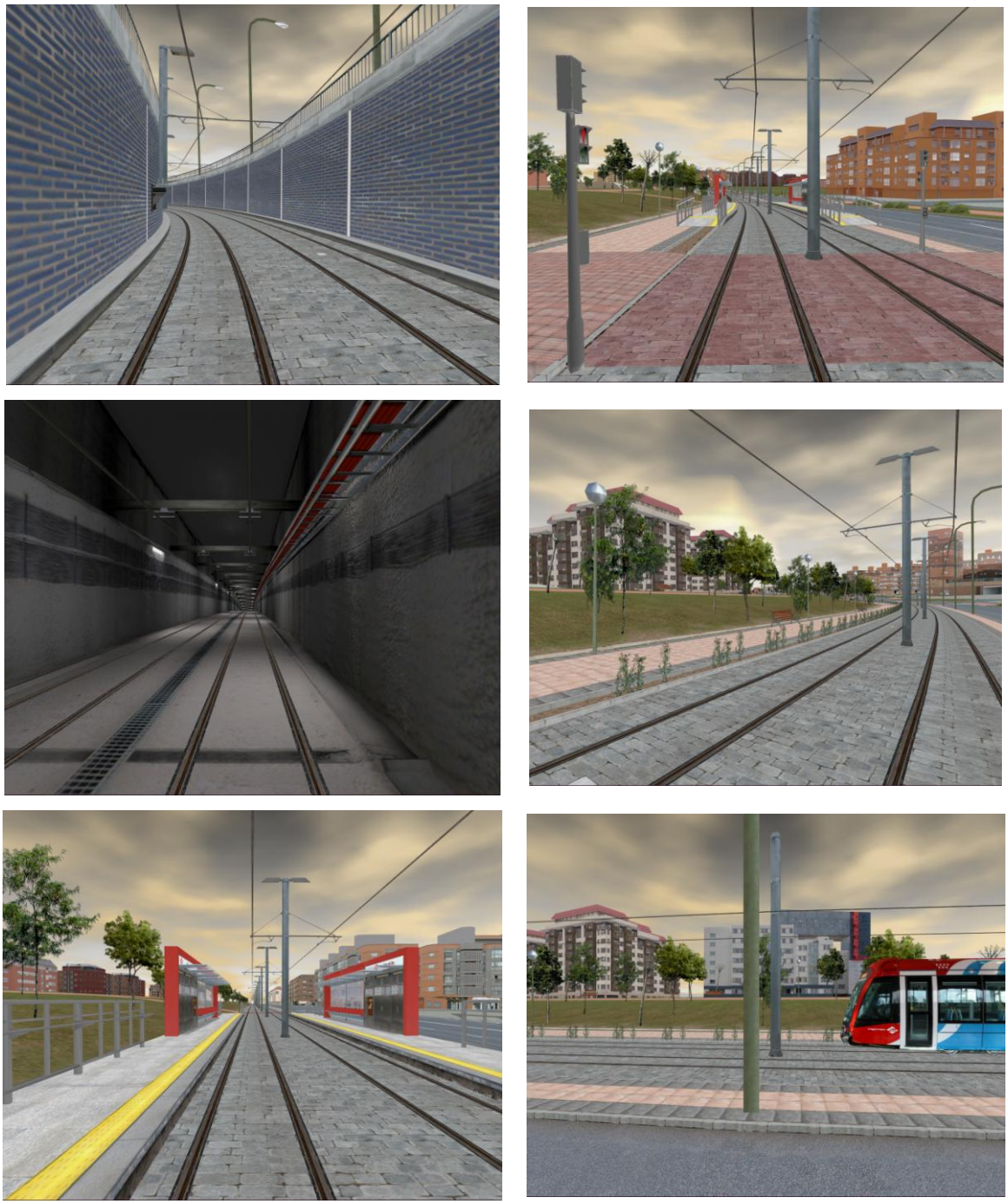


Figura 5.76.Escenarios de la línea ML1 de Metro Ligero de Madrid.

En 2009 CITEF comenzó el Simulador de Formación en Conducción y Averías de Metro de Santiago de Chile. El objetivo fundamental del Simulador era proporcionar una herramienta destinada a la formación del personal de Operaciones de METRO S.A. en la conducción, el tratamiento de averías, el manejo de los equipos embarcados y la actuación de acuerdo a los protocolos definidos y regulados en la explotación, tanto sobre vías principales (vías 1 y 2 de cada línea), como sobre vías secundarias (vías con designación A, B, C,……Z que se encuentra en túneles de enlace, zonas de maniobras, depósitos, cocheras y terminales) para cada una de las tecnologías de material rodante simuladas y para cada una de las líneas objeto de simulación. En

este proyecto la Tecnología Modular colaboró en la realización de los entornos virtuales de Linea 1 de metro de Santiago de Chile (Figura 5.77).

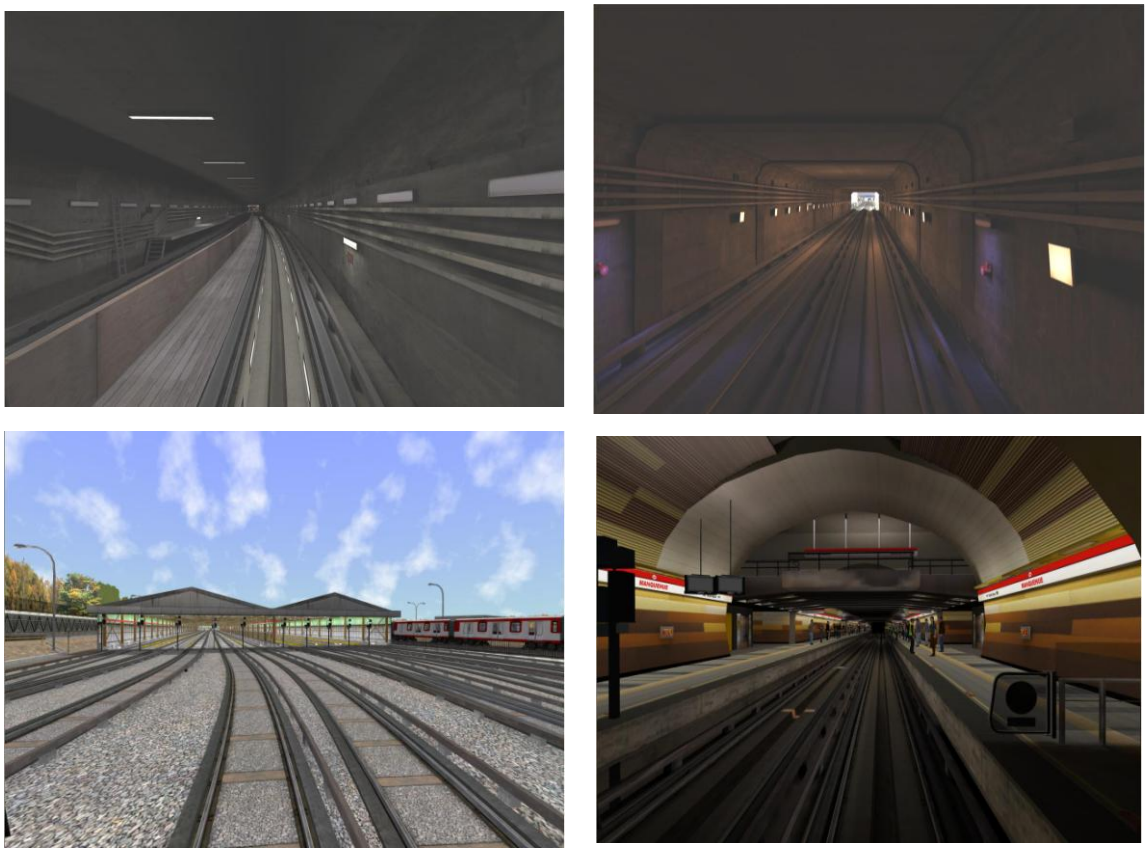


Figura 5.77.Escenarios desarrollados para la Linea 1 de Metros de Santiago de Chile.

Este es el primer proyecto en el que se aplica la generación modular con shaders. Los resultados de la metodología empleada en ese momento se describieron en los artículos presentados en GAMEON 2009[328], UKSIM 2010[329] y The Industrial Simulation Conference 2010, ISC'10 [330].

[328] Tovar, C., Jimena de Dios, G., Cabanellas, J.Mª. Modular Technology in the generation of large virtual environments. 10th Internacional Conference on Intelligent Games and Simulation, GAMEON 2009.November 2009, Düsseldorf, Germany.

[329] Tovar, C., Jimena de Dios, G., Cabanellas, J.Mª, Zoido, C.. Modular Technology in the modelling of large virtual environments in driving simulators. Internacional Conference on Computer Modelling and Simulation. UKSIM 2010. Cambridge, England.

[330] Tovar, C., Jimena de Dios, G., Cabanellas, J.Mª. J. Modular Technology: a methodology for the creation of virtual environments in terrain driving simulators. The Industrial Simulation Conference. ISC'10. June 2010. Budapest, Hungary.

6. CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN.

6.1 CONCLUSIONES.

Tras un exhaustivo estudio de la problemática asociada a la generación de grandes entornos virtuales para simulaciones de conducción terrestre guiada bajo PC destinados al aprendizaje y entrenamiento de conductores, la presente Tesis concluye que es necesario disponer de una metodología que facilite dicha generación. Dicha facilitación debe priorizar los objetivos del entrenamiento de conducción, lo cual implica:

- satisfacer los requisitos funcionales de la conducción.
- cumplir las necesidades del motor gráfico, para ello:
    - se han de minimizar los tiempos de carga y la memoria de almacenamiento;
    - se han de conseguir las velocidades de refresco necesarias ;
- generar un flujo de trabajo continuo entre las diversas fases constructivas del entorno que garantice la eficiencia:
    - automatizar en la mayor medida posible todas las tareas que han de llevarse a cabo;
    - centralizar el flujo de información entre los diversos subsistemas que integran la simulación;
    - garantizar en todo momento la coherencia de dicha información y su perdurabilidad en el tiempo;
- proporcionar herramientas que garanticen de una manera rápida e intuitiva la generación de nuevos escenarios, permitiendo la reutilización de otros escenarios ya creados;
- versatilidad a la hora de satisfacer las necesidades del cliente, esto implica:
    - la posibilidad de ofrecer una generación progresiva en la que el grado de detalle final se establecerá en función de la inversión económica que desea realizar el cliente;
    - la posibilidad de readaptación de los escenarios generados a los últimos avances del hardware, con el fin de ofrecer siempre la solución tecnológica más avanzada con un mínimo esfuerzo.

La presente Tesis ha conseguido todos estos requisitos mediante el desarrollo de la Tecnología Modular, una metodología que ofrece como resultado final la automatización optimizada en la generación de grandes entornos virtuales para simuladores de conducción terrestre guiada bajo PC. Para ello se ha abordado el ciclo completo de generación de un entorno virtual, aportando soluciones optimizadas a cada una de las fases de este proceso constructivo. Estas fases han sido sintetizadas por esta Tesis en tres grupos:

- **Fase1: Modelado Conceptual.**

En esta fase esta Tesis ha explotado las características repetitivas de este tipo de entornos de una manera hasta ahora no realizada, aunando las ventajas del modelado ad hoc y del modelado por losetas. Para ello el entorno se sintetiza en base a un conjunto de patrones y módulos. Los patrones, generados a partir de mallas ad hoc, obeden a una serie de reglas de generación que permiten su automatización . Los módulos, caracterizados por la definición de una serie de perfiles transversales y de texturas, se ensamblan e instancian a lo largo de una serie de líneas directrices generando las zonas de repetibilidad. Gracias a este sistema de patrones se consigue:

- desarrollar un sistema escalable: con un número finito de patrones es posible generar un número infinito de entornos.
- disminuir los tiempos de carga y la memoria de almacenamiento;
- aportar mayor flexibilidad constructiva y realismo que los sistemas de losetas empleados hasta el momento.
- disminuir las tareas manuales a realizar.
- facilitar las labores de creación y edición de entornos, algo fundamental para el usuario final de estas simulaciones;

Otra aportación de la Tesis ha consistido en un conjunto de herramientas que permiten la creación de dichas familias modulares, su edición y combinación. Estas herramientas, permiten de una manera cómoda e intuitiva sin necesidad de conocimientos infografistas, la generación optimizada de modelos 3D del escenario. El almacenamiento estructurado de estos modelos en base a la definición de un conjunto de perfiles transversales característicos, minimiza la memoria requerida y maximiza la reutilización de los mismos, ya que tanto los perfiles como las texturas asociados a cada familia pueden ser modificados y combinados fácilmente para generar nuevas familias modulares.

- **Fase 2: Modelado de Datos.**

Tras estudiar las posibles aportaciones que los CAD y Sistemas de Información Geográfica ofrecen en este ámbito y evaluando sus carencias, esta Tesis ha desarrollado un sistema consistente en varios criterios de organización, corrección y almacenamiento de la información de partida, particularizado a las necesidades de una simulación de conducción terrestre. Dicho sistema permite por un lado la construcción fácilmente escalable y editable de entornos que cumplan las normativas circulatorias vigentes y por otro lado garantiza un óptimo flujo de la información entre los diversos subsistemas integrantes de la simulación. Este sistema se ha visto plasmado por un lado en diversas herramientas que permiten la creación y edición del entorno de manera rápida y eficaz. El software desarrollado permite así, construir un sistema de información geográfica a medida por medio del uso de un conjunto de herramientas que contribuyen a la minimización de tiempo y esfuerzo.
Por otro lado este sistema se ha visto plasmado en la elaboración de diversos protocolos de comunicación entre los diversos subsistemas integrantes de la simulación que garantizan un correcto flujo de la información entre los mismos.
Finalmente este sistema ofrece la posibilidad de conectarse a otras herramientas comerciales[331] para aquellos usuarios finales que lo deseen, con el fin de poder ampliar la funcionalidad del mismo.

[331] http://usa.autodesk.com/ [Última consulta: 7 Enero 2013]

- **Fase 3: Modelado Físico.**

  En esta fase, la presente Tesis aporta una serie de Algoritmos de Posicionamiento modular que garantizan el correcto acoplamiento de los módulos a lo largo de una serie de líneas directrices. Dichas líneas son creadas a partir de la definición de las trayectorias circulatorias, lo que ha permitido a su vez, la definición por parte de esta Tesis de un sistema de niveles de detalle discretos pseudo-variantes con el punto de vista, que permite optimizar la carga geométrica del escenario mediante la definición en tiempo de precarga del nivel de detalle de cada módulo, en función de su distancia transversal a la trayectoria.
  Por otro lado, esta Tesis aporta un grafo de la escena, que gracias al empleo de la Tecnología Modular, facilita las labores de revisión y edición del mismo. A su vez, este grafo permite dotar de comportamiento a todos aquéllos elementos del entorno que lo necesiten, como aparatos de vías y regulación semafórica.
  Por último, esta Tesis garantiza la consecución de las velocidades de refresco necesarias en este tipo de simulaciones mediante la optimización para cada entorno virtual de los diversos parámetros intervinientes en el grafo de la escena. Variables como las distancias de LOD y el tamaño del módulo se adaptan a los requisitos de cada escenario. La gran versatilidad ofrecida por la Tecnología Modular permite variar la definición de la familia modificando tanto el conjunto de perfiles transversales que la definen como el tamaño básico del módulo, de manera que el nivel de instanciación de cada entorno puede ser graduado como se desee pudiendo degenerar en caso necesario en una generación clásica ad hoc (por bloques de mallas).

6.2 FUTURAS LÍNEAS DE INVESTIGACIÓN.

Las futuras líneas de investigación son numerosas debido tanto al gran desarrollo que está adquiriendo el mundo de la simulación en los últimos años como al posible empleo de la Tecnología Modular para la creación de entornos virtuales destinados a otros fines. No obstante cabe destacar las siguientes:

- Creación de una IDE [332] (Infraestructura de Datos Espaciales) que permita centralizar las fuentes de información disponibles para la generación de simulaciones de conducción terrestre. Sería interesante poder visualizar vía web, a través de esta IDE, los entornos virtuales generados, aprovechando las facilidades de transmisión y almacenamiento que puede ofrecer el empleo de este sistema modular. A su vez, esta IDE podría emplearse como herramienta para llevar a cabo la gestión del mantenimiento de la infraestructura ferroviaria.
- Conexión del Módulo de ajuste Geométrico a otras posibles aplicaciones SIG como pueden ser ArcGis [333] , Geomedia [334] o gvSIG [335].
- Incorporación de nuevos shaders que permiten aumentar la versatilidad constructiva y el realismo. Por ejemplo:
  - Nuevos shaders para la generación automática de terrenos.
  - Estudiar el posible empleo de shaders geométricos que permitan una completa generación en tiempo de ejecución de la geometría de los módulos.

[332] http://www.idee.es [Última consulta: 7 Enero 2013]

[333] http://www.esri.com/software/arcgis/index.html [Última consulta: 7 Enero 2013]

[334] http://www.intergraph.com/sgi/products/ [Última consulta: 7 Enero 2013]

[335] http://www.gvsig.org/ [Última consulta: 7 Enero 2013]

- Estudiar el posible empleo de la Tecnología Modular para entornos de conducción terrestre no guiada, modificando la forma del módulo y el sistema de posicionamiento.
- Automatizar la optimización de los parámetros intervinientes en el grafo de la escena.
- Automatización en la determinación de distancias de los diferentes niveles de detalle mediante análisis de visibilidad.
- Aplicación de la Tecnología Modular sobre nuevas plataformas: teléfonos móviles, tablets,etc.

NEXO 1. NOMENCLATURA DEL MÓDULO.

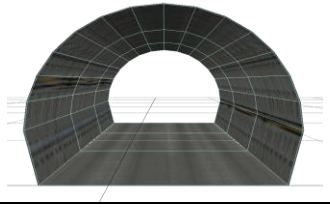
La nomenclatura empleada por la Tecnología Modular para designar los elementos 3d a insertar en el grafo de la escena (excepto elementos de señalización, que debido a su complejidad requieren de una nomenclatura propia que se explica en el Anexo 7) es la siguiente:

- LOD: Dos siglas designan el nivel de detalle con el que se ha modelado el objeto:N1,N2,N3..etc.
- PROXIMIDAD: Una sigla designa el nivel de distancia del objeto a la línea directriz que lo posiciona: 0,1,2,3..etc.
- ID FAMILIA: Dos siglas designan el nombren de la familia.
- ID SUBFAMILIA: Dos siglas designan el nombre de la subfamilia.
- TALUDES: Dos siglas designan el desnivel de alturas que se produce en dirección longitudinal a lo largo del objeto.
- ID TEXTURA: Dos siglas designan el tipo de textura.
- NUMERO DE VIAS ABARCADAS: Dos siglas designan el número de vías abarcadas por el objeto.
- ID PROYECTO: Una sigla designa el tipo de proyecto al que pertenece el modelo: ferroviario, urbano o genérico.
- SECUENCIA HORIZONTAL: I,F,C . En el caso de familias que exigen en sentido longitudinal una secuencia de colocación determinada, esta sigla designa si el objeto es de inicio, mitad o fin de esta secuencia.
- X: se deja un campo vacío por si surge la necesidad de completar esta nomenclatura.
- SECUENCIA_TRANSVERSAL:I,D,C. En el caso de familias que exigen en sentido transversal una secuencia de colocación, esta sigla designa si el objeto es del lado derecho, centrado o izquierdo en esta secuencia.
- LADO: I,D,C. Dos siglas designan el lado en el que se coloca el objeto respecto de la línea directriz que lo posiciona: derecho,centrado o izquierdo.
- ANGULO: Un sigla designa si el objeto está girado a derechas ,izquierdas o es recto (D,I,R) y las siguientes siglas designan el ángulo girado o la longitud básica del módulo, dependiendo de si se trata un módulo curvo o recto.

A continuación se muestran diversos ejemplos de familias y la nomenclatura empleada.

TUNELES

TUNEL ELIPSE

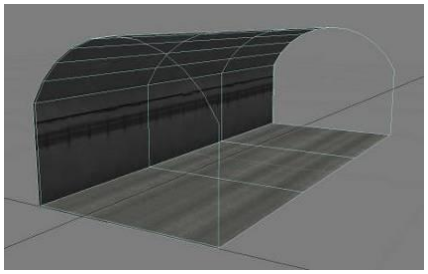
LOD	PROXIMIDAD	ID FAMILIA	ID SUBFAMILIA	TALUDES	TEXTURA	Nº VIAS	ID PROYECTO	S.H	x	S.V	ANGULO
N1	0	TD	AAA	00	TN	2	M	C	X	C	R200
											

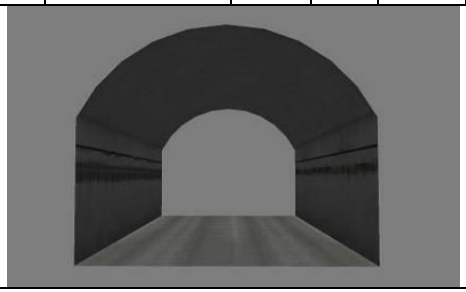
TUNEL INDIVIDUAL BELGA

LOD	PROXIMIDAD	ID FAMILIA	ID SUBFAMILIA	TALUDES	TEXTURA	Nº VIAS	ID PROYECTO	S.H	x	S.V	ANGULO
N1	0	TS	AAB	00	TN	1	M	C	X	C	R200
											

TUNEL BELGA

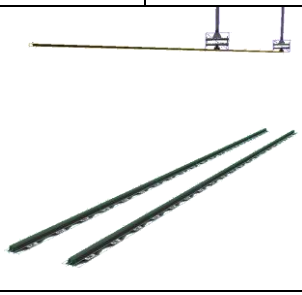
LOD	PROXIMIDAD	ID FAMILIA	ID SUBFAMILIA	TALUDES	TEXTURA	Nº VIAS	ID PROYECTO	S.H	x	S.V	ANGULO
N1	0	TD	AAC	00	TN	2	M	C	X	C	R200





VIA

LOD	PROXIMIDAD	ID FAMILIA	ID SUBFAMILIA	TALUDES	TEXTURA	Nº VIAS	ID PROYECTO	S.H	x	S.V	ANGULO
N1	0	VI	AAA	00	TN	1	M	C	X	C	R200



ANEXO 2. PROTOCOLOS DE COMUNICACIÓN PARA UNA SIMULACIÓN DE CONDUCCIÓN.

1.INTRODUCCIÓN.

El objetivo de este Protocolo es definir el flujo de información gobernado por la Tecnología Modular entre los diversos subsistemas involucrados en las simulaciones de conducción del CITEF: motor gráfico, simulación ferroviaria, tráfico rodado y de peatones (MicroTraff), y sector infografista. En dichas simulaciones podrá coexistir tráfico ferroviario y urbano, como es el caso de los simuladores desarrollados para las líneas de Metro Ligero de Madrid.

Se remarcarán los puntos más importantes y conflictivos que se tendrán que decidir al comienzo del proyecto en cuanto a la configuración de la infraestructura del simulador y se definirán las estructuras de datos a generar por la Tecnología Modular para cada subsistema.

2.CONDICIONES FIJAS O LIMITACIONES DE PROYECTO.

Las condiciones que se muestran a continuación una vez fijadas no se deben cambiar a lo largo del proyecto. Si se encuentran casos en los que no se puedan cumplir estas limitaciones, entonces se tendrán que discutir esos casos con las otras partes que se pueden ver afectadas por el problema:

- Tolerancia del reescalado visual del tren: Con el fin de que la simulación sea lo más real posible, el tren no podrá escalarse un valor superior a esta tolerancia. Lo mismo ocurrirá con las longitudes de las trayectorias, con el fin de que la diferencia entre las medidas que se representan en el visual no discrepen en exceso con las que aparecen en los planos de señalización.

- Dimensiones de las estaciones: el tamaño de la estación deberá quedar prefijado al comienzo del proyecto para que su dimensión sea razonable para todos los grupos. Habrá que tener en cuenta el tamaño del tren para comprobar que el tamaño que tiene la estación es suficiente para que entre el tren.

- Pendientes máximas: ya que los trenes no pueden superar grandes pendientes, las pendientes de las trayectorias deberían estar dentro de unos límites acordados, dependiendo del proyecto y del tren a simular.

- Algoritmo de cálculo de la poligonal y otras curvas: los puntos de la poligonal y los puntos intermedios son datos de la configuración del escenario. Por tanto, tanto la poligonal que define las trayectorias como el método de obtención de los puntos intermedios son información de uso común.

3.DATOS DE PARTIDA ACORDADOS.

Debido a la existencia de incongruencias entre las fuentes de información a manejar por los diversos grupos involucrados en la simulación, se han de adoptar soluciones de compromiso en aquellos puntos que puedan afectar a más de un grupo:

- Tamaño del tren: tendrá que quedar perfectamente definido antes de fijar las longitudes de la vía.
- Tamaño y posición de las estaciones: es importante acordar dónde van a ir las estaciones y las particularidades que pueden presentar cada una de ellas.
- Posición de nodos: se puedan dar situaciones que impliquen la necesidad de imponer la posición de algunos nodos.
- Longitud de las secciones: debido a situaciones particulares puede ser necesario fijar de antemano el tamaño de algunas secciones.
- Posición de elementos de vía: se decidirá que elementos de vía se deben simular y si hay alguno que tiene que estar en una situación especial.
- Posición y distancia entre espejos: se elegirán que espejos se van a simular o que criterio general se va a seguir. Puede ser importante que la distancia entre unos espejos sea fijada de antemano.
- Túneles: habrá que decidir qué túneles se simulan y si han de incorporar algún elemento o señalización especial.

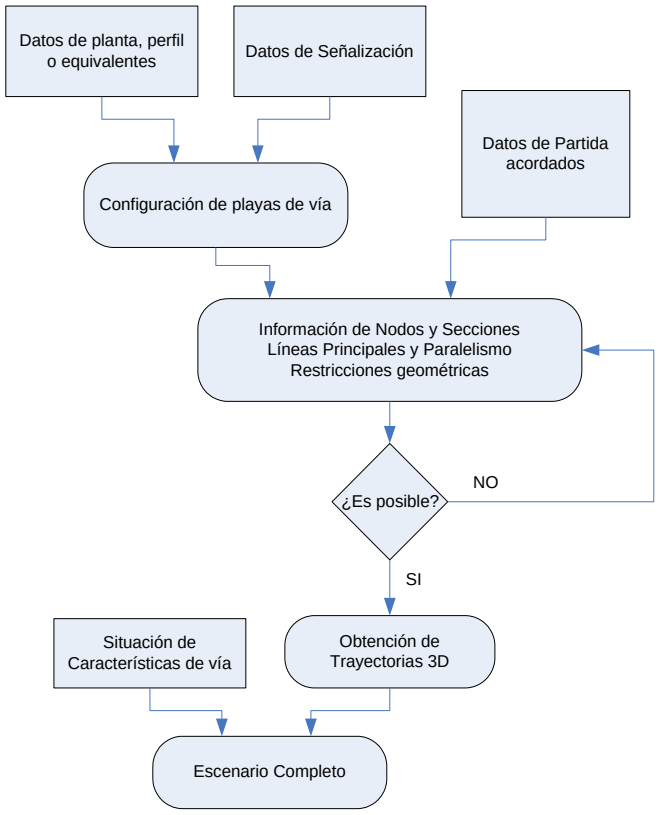


Figura A2.1 Esquema de trabajo.

4.DEFINICIONES PREVIAS.

- Nodo. Corazón de Aguja, desvío o cualquier instalación en vía que suponga una bifurcación. Es un punto de conexión entre vías. Se puede definir un nodo virtual por conveniencia, en ese caso es un punto de una vía sin ninguna correspondencia con una instalación o característica de la vía. El nodo se coloca en el punto exacto en el que la traza de las vías se bifurca. Los nodos terminales se corresponden con los inicios-fines de vía: toperas o similares.

- Sección. La zona de una vía entre dos nodos.
- Línea. Una sucesión de secciones. En ocasiones se identifica con un número de vía. Las líneas suelen tener una cierta relación de paralelismo entre ellas.
- Línea Principal (LP). Una línea escogida arbitrariamente que sirve para garantizar el correcto posicionamiento relativo de elementos que se encuentran en líneas distintas, ya que los puntos con el mismo PK en la línea Principal (PK Absoluto) están "enfrente". Suele designarse vía 1 como la línea principal.
- Posición. Punto sobre la infraestructura que sirve para localizar instalaciones, características de vía, modelos de entorno 3D, etc. Cuando una característica tiene longitud se distinguen puntos iniciales y finales.
- Poligonal. Es el conjunto de puntos por los que pasa la vía. Estos puntos se unen por segmentos y sobre la línea que se forma se construye el entorno en 3D.
- PK Absoluto (PKabs). Identificador de posición para los elementos de vía asociado a los planos de documentación. El PK absoluto lo define la línea principal, en metros.
- PK. Medida de posicionamiento para los elementos de vía. El PK se corresponde de forma exacta con el que se define para los sistemas de señalización (en metros).
- PKr (PK relativo). Medida de posicionamiento relativo para los elementos de vía. Su definición implica determinar el identificador del elemento de referencia y el PK respecto de éste. Si la distancia es negativa el punto de referencia se encuentra a la derecha del punto situado.
- Espacio Visual (E). Medida de posicionamiento para los elementos de vía. El espacio de un punto es la distancia medida sobre la traza de vía en el entorno en 3D desde el punto en cuestión al nodo izquierdo de la sección (en metros).
- Espacio Visual Relativo (ER). Medida de posicionamiento para los elementos de vía. Su definición implica determinar el identificador del elemento de referencia y la distancia medida sobre la traza de vía en el entorno 3D. Si la distancia es negativa el elemento de referencia se encuentra a la derecha. El ER puede medirse respecto del punto izquierdo o derecho de una característica de vía con longitud.
- Número de vía. Es un número de información de referencia. Su función es orientar sobre la situación de la vía.
- Número de posición. Suele coincidir con el número de vía. Es un número que indica la posición lateral (izquierda y derecha) de la sección respecto de la LP.

5.INFORMACIÓN DE PARTIDA DEL ENTORNO.

A continuación se definen los archivos de configuración requeridos para la generación de un entorno ferroviario y urbano.

5.1.INFORMACIÓN DE PARTIDA DEL ENTORNO FERROVIARIO.

La información de partida para el entorno ferroviario se introducirá a través de la aplicación GENENT, herramienta desarrollada por la Tecnología Modular para el CITEF. Dicha información queda sintetizada en los siguientes archivos de configuración:

1. cfg_conectividades:

- m_seccion_ini
- m_seccion_fin
- m_nodo_mov
- m_nombre_nodo
- m_tipo_nodo
- m_estado

2. cfg_lineas:
   - m_nombre_seccion
   - m_posicion
   - m_pki_abs_linea
   - m_pki
   - m_pkf
   - m_max_speed

3. cfg_radios:
   - m_nombre_seccion
   - m_pki
   - m_pkf
   - m_radio
   - m_longitud

4. cfg_gradientes
   a. m_nombre_seccion
   b. m_pki
   c. m_pkf
   d. m_gradientes

5. cfg_intervalos:
   - m_nombre_linea
   - m_id_linea
   - m_tipo_intervalo
   - m_pki_abs
   - m_pkf_abs
   - m_nombre_linea_pral
   - m_distancia_paralelismo

6. cfg_lineas_prales
   - *m_nombre_linea*
   - *m_id_linea*
   - *m_coorx*
   - *m_coory*
   - *m_coorz*
   - *m_alpha*
   - *m_beta*
   - *m_gamma*

7. cfg_lineas_secundarias
   - *m_nombre_linea*
   - *m_id_linea*
   - *m_coorx*
   - *m_coory*

- *m_coorz*
- *m_alpha*
- *m_beta*
- *m_gamma*

8. cfg_entornos
   - *m_id_entorno*
   - *m_nombre_linea*
   - *m_proximidad*
   - *m_lado*
   - *m_familia*
   - *m_subfamilia*
   - *m_Tipo_elemento_referencia*
   - *m_Extremo_de_referencia*
   - *m_tipo_coordenada*
   - *m_coorx*
   - *m_coory*
   - *m_coorz*
   - *m_alpha*
   - *m_beta*
   - *m_gamma*
   - *m_pki*
   - *m_si*
   - *m_Tipo_elemento_referenciaf*
   - *m_Nombre_elemento_referenciaf*
   - *m_Extremo_de_referenciaf*
   - *m_tipo_coordenadaf*
   - *m_coorxf*
   - *m_cooryf*
   - *m_coorzf*
   - *m_alphaf*
   - *m_betaf*
   - *m_gammaf*
   - *m_pkf*
   - m_sf

9. cfg_señalizacion:
   - *m_id_señal*
   - *m_Nombre_señal*
   - *m_Nombre_linea*
   - *m_lado*
   - *m_Sentido*
   - *m_Tipo_elemento_referencia*
   - *m_Nombre_elemento_referencia*
   - *m_Extremo_de_referencia*
   - *m_Tipo_coordenada*
   - *m_coorx*
   - *m_coory*
   - *m_coorz*
   - *m_alpha*
   - *m_beta*
   - *m_gamma*

- *m_PK*
- *m_S*

10. cfg_consolidacion:
    - *m_nombre_linea*
    - *m_id_linea*
    - *m_coorx*
    - *m_coory*
    - *m_coorz*
    - *m_alpha*
    - *m_beta*
    - *m_gamma*
    - *m_pkabs*

11. cgf_intervalos_ctes: es el listado de intervalos que no pueden ser consolidados
    - *m_nombre_linea*
    - *m_id_linea*
    - *m_pki_abs*
    - *m_pkf_abs*

12. cfg_intervalos_planta: contiene la lista de intervalos que definen las zonas de generación con radios y con splines para la consolidación
    - *m_nombre_linea*
    - *m_id_linea*
    - *m_pki_abs*
    - *m_pkf_abs*
    - m_tipo_intervalo

5.2.INFORMACIÓN DE PARTIDA DEL ENTORNO URBANO.

La información de partida para el entorno urbano se introducirá a través de URBEDIT, herramienta desarrollada por la Tecnología Modular para el CITEF. A través de URBEDIT es posible la introducción de todos los nodos de la red urbana, ya sea de manera ficticia o empleando alguna plantilla de Autocad como base, formato en el que suele darse la información para los simuladores desarrollados para el CITEF.

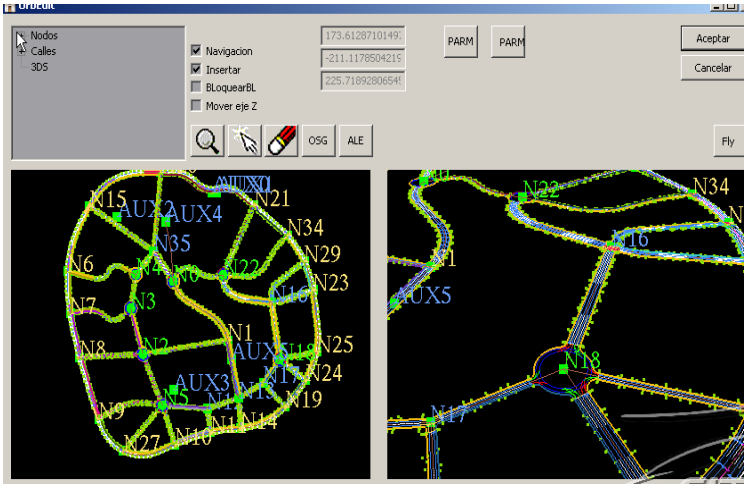


Figura A2.2 URBEDIT

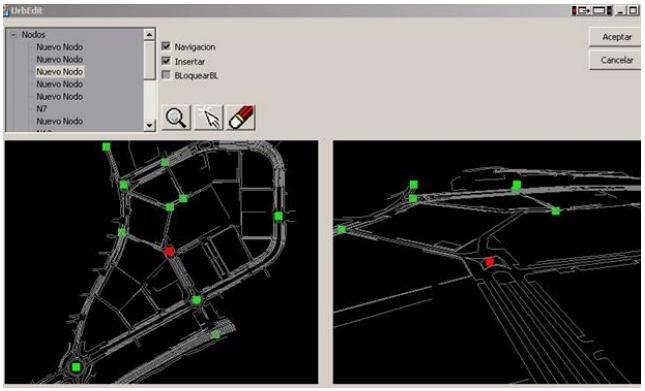


Figura A2.3 Interfaz de entrada de nodos de Urbedit.

A partir de la definición de nodos es posible definir las calles, indicando el número de carriles de la misma, posicionamiento transversal, ancho y sentido.

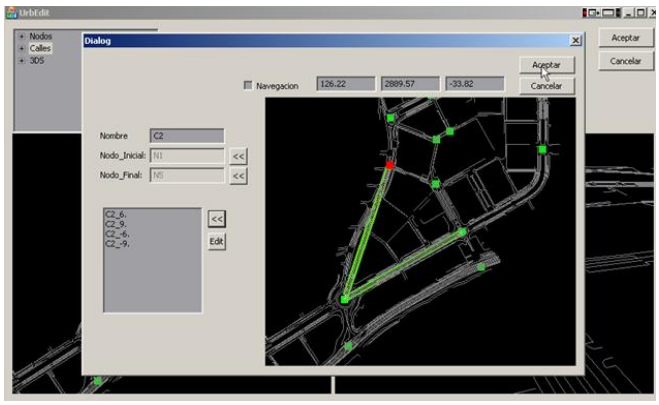


Figura A2.4 Interfaz de entrada de calles de Urbedit.

Una vez introducidos los carriles es posible editar su geometría mediante desplazamiento de los vértices de su poligonal de control así como los parámetros que definen cada tipo de intersección.

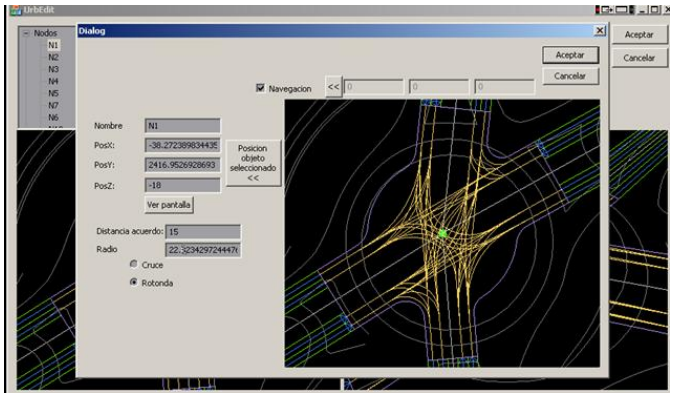


Figura A2.5 Interfaz de edición de Urbedit : modificación de la poligonal de control de un carril y del radio de una rotonda.

Dicha información se sintetiza en los siguientes archivos de configuración:

- cfg_ciudad_nodos_constructivos.txt: contiene la información de los nodos de la ciudad:
  - Nombre del nodo.
  - Identificador del nodo.

    - Coordenadas del nodo: x, y, z, alfa, beta, gamma.

- cfg_ciudad_conectividades.txt: define las conectividades entre nodos:
    - Curva constructiva de entrada.
    - Curva constructiva de salida.
    - Identificador del nodo.
    - Tipo de nodo.

- cfg_ciudad_curvasconstructivas.txt: define las curvas constructivas que servirán como base para generar la geometría de la red urbana:
    - Identificador de la curva constructiva.
    - Secuencia de nodos que la definen.
    - Línea ficticia asociada, en casa de existir alguna relación de paralelismo con el entorno ferroviario.
    - Distancia de paralelismo a la línea ficticia.

- cfg_ciudad_carriles.txt: define la información relacionada con carriles:
    - Identificador.
    - Posición.
    - Ancho.
    - Tipo.

- cfg_ciudad_calles.txt: define la información de calles.
    - Identificador de la calle.
    - Secuencia de nodos que la definen.

- cfg_ciudad_radios.txt: define los radios de las curvas constructivas:
    - Nombre de la Curva Constructiva.
    - Pki.
    - Pkf.
    - Radio.
    - Longitud.

- cfg_ciudad_gradientes.txt: define los gradientes de las curvas constructivas:
    - Nombre de la Curva Constructiva.
    - Pki.
    - Pkf.
    - Gradiente.

- cfg_ciudad_Intervalos.txt: define los intervalos de paralelismo, conexión e independencia de los carriles:
    - Nombre del carril.
    - Tipo de intervalo.
    - Pki abs.
    - Pkf abs.
    - Nombre de la Curva Constructiva.
    - Distancia de paralelismo.

- cfg_ciudad_radios_acuerdo.txt: define los radios de acuerdo de las intersecciones:
    - Identificador del acuerdo.
    - Radio de acuerdo.

- cfg_ciudad_rotondas.txt: define las características de las rotondas:
    - Identificador de la rotonda.
    - Radio externo.
    - Radio interno.
    - Número de carriles de las rotondas.

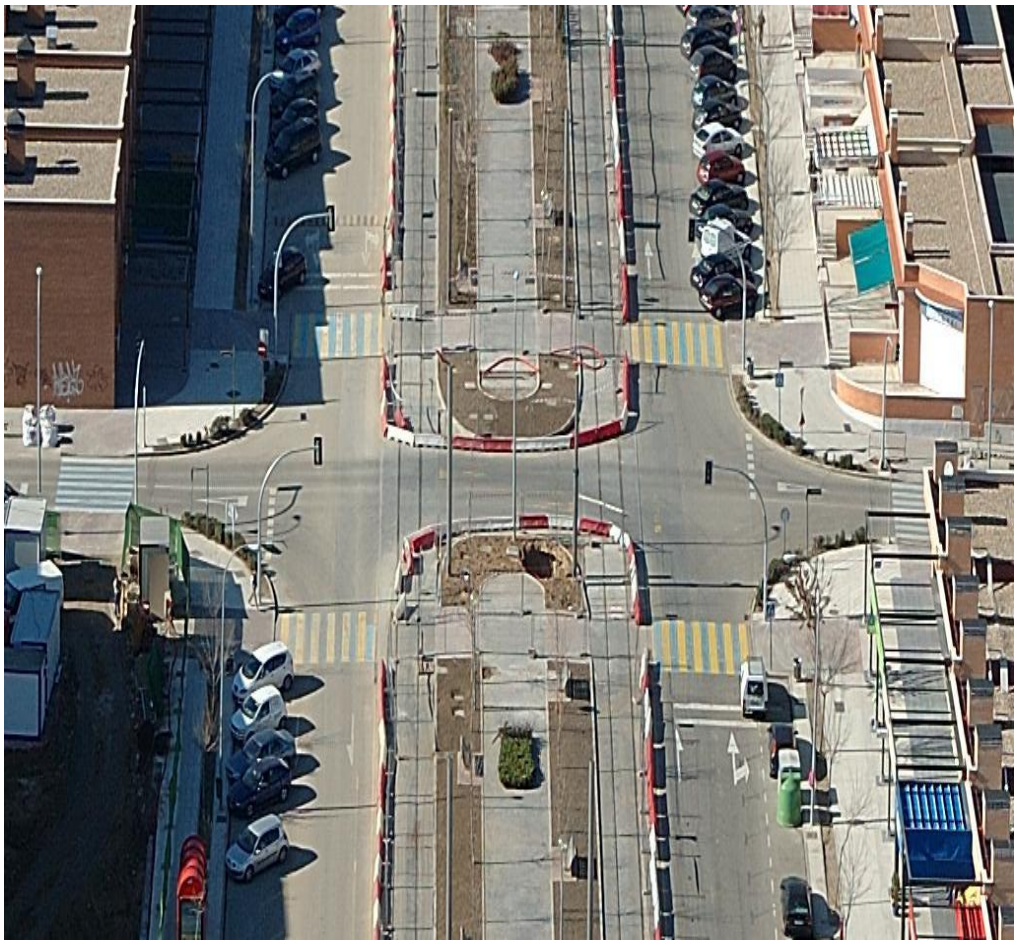
- cfg_ciudad_entornos.txt: define la información del entorno urbano a colocar:
    - Id del entorno.
    - Nombre del carril.
    - Proximidad.
    - Lado.
    - Nombre de la familia.
    - Nombre de la subfamilia.
    - Tipo de elemento de referencia inicial.
    - Extremo de referencia inicial.
    - Tipo de coordenada inicial.
    - Coordenada X inicial.
    - Coordenada Y inicial.
    - Coordenada Z inicial.
    - Coordenada Alpha inicial.
    - Coordenada Beta inicial.
    - Coordenada Gamma inicial.
    - Pk inicial.
    - S inicial.
    - Tipo de elemento de referencia final.
    - Nombre del elemento de referencia final.
    - Extremo de referencia final.
    - Tipo de coordenada final.
    - Coordenada X final.
    - Coordenada Y final.
    - Coordenada Z final.
    - Coordenada Alpha final.
    - Coordenada Beta final
    - Coordenada Gamma final
    - Pk final.
    - S final.

- cfg_ciudad_marcas_viales.txt: define el posicionamiento y tipo de las marcas viales a colocar.
    - Identificador de la marca vial.
    - Tipo de elemento de referencia.
    - Identificador del elemento de referencia.
    - Posición inicial.
    - Posición final.
    - Tipo de marca vial.
    - Pk inicio.
    - Pk fin.

- cfg_ciudad_sennalizacion.txt: define la regulación semafórica a colocar.
    - Id de la señal.
    - Nombre de la señal.

- Nombre del carril.
- Lado.
- Sentido.
- Tipo de elemento de referencia.
- Nombre del elemento de referencia.
- Extremo de referencia.
- Tipo de coordenada.
- Coordenada X.
- Coordenada Y.
- Coordenada Z.
- Coordenada Alpha.
- Coordenada Beta.
- Coordenada Gamma.
- PK.
- S.

Para la introducción de la señalización urbana se han generado plantillas que permiten añadir de manera automática señales en intersecciones y ramales pertenecientes a entornos ficticios. Para cada tipo de intersección (cruce, rotonda, incorporación, etc.) existe una diversidad de configuraciones, según el número de ramales que confluyen en ella, si tienen mediana o no, si hay vías de tranvía o no, etc. Por ejemplo, para una intersección de tipo cruce de cuatro ramales: dos principales con dos carriles en cada sentido, mediana y tranvía y dos secundarios con un carril en cada sentido, la figura siguiente muestra la señalización a emplear:



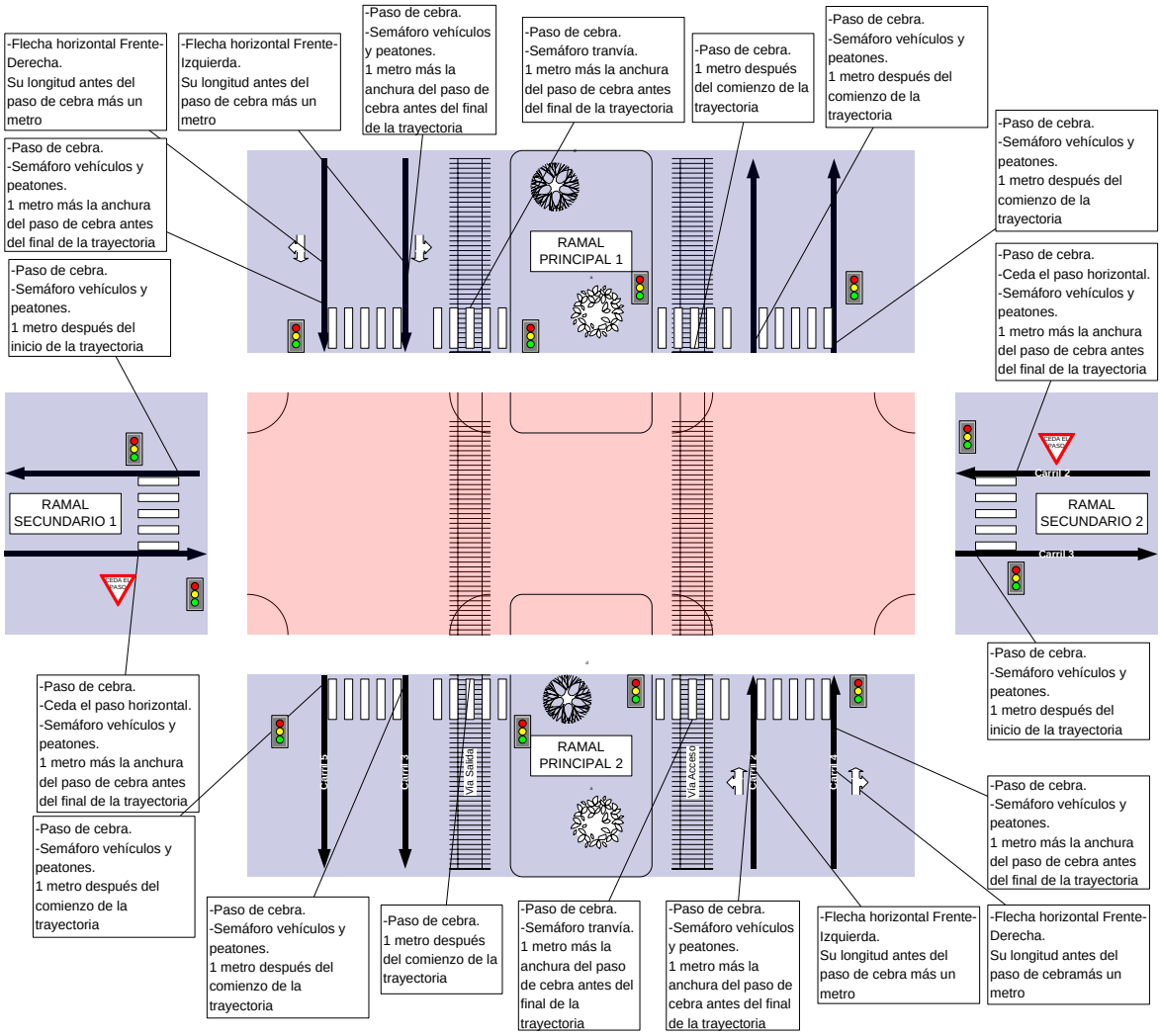


Figura A2.6 Esquema de señalización para una intersección de tipo cruce[336].

La señalización a colocar en los diferentes ramales será la siguiente:

- Señalización en ramales principales (ambos ramales tienen la misma señalización):

Se designa con Lp a la longitud de un paso de cebra (supuesta igual para todos los pasos de cebra), con Lf a la longitud de una flecha de preselección de carril (supuesta igual para todas las flechas), con Lc a la longitud de un ceda el paso (supuesta igual para todos los ceda el pasos. En el Apéndice Grupos Semafóricos que se da al final de este anexo se detallan los distintos tipos de grupos semafóricos usados en el ejemplo (cada grupo semafórico se refiere a un mástil con una serie de carcasas y focos).

[336] Las líneas gruesas con flecha indican cada carril del tráfico rodado y su sentido. Dos líneas finas paralelas con líneas transversales indican vía de tranvía. Las anotaciones recuadradas contienen las señales y el Pk de su línea de detención. Están unidas con una línea fina al lugar aproximado de su línea de detención.
Los símbolos de las señales se han colocado en el lugar aproximado en el que ha de colocarse en el visual

Carril / Vía	Pk de la línea de detención	Señal	Localización y orientación del modelo 3D.
Carril 2	1+ Lp metros antes del final de la trayectoria	MARCA VIAL DE PASO DE CEBRA	
Carril 4	1+ Lp metros antes del final de la trayectoria	MARCA VIAL DE PASO DE CEBRA	
Carril 2	1+ Lp metros antes del final de la trayectoria	GRUPO SEMAFÓRICO TIPO 2 (vehículos y peatones)	Mediana, junto a la vía de tranvía, al final del paso de cebra (sentido Carril 2). La carcasa de vehículos orientada en sentido contrario al Carril 2 y la de peatones mirando a la acera opuesta
Carril 4	1+ Lp metros antes del final de la trayectoria	GRUPO SEMAFÓRICO TIPO 1 (vehículos y peatones)	Acera derecha, al final del paso de cebra (sentido Carril 4). Las carcasas de vehículos orientadas en sentido contrario al Carril 4 y la de peatones mirando a la acera opuesta
Carril 2	2+ Lp+ Lf metros antes del final de la trayectoria	MARCA VIAL DE FLECHA DE PRESELECCIÓN DE CARRIL DE FRENTE E IZQUIERDA	
Carril 4	2+ Lp+ Lf metros antes del final de la trayectoria	MARCA VIAL DE FLECHA DE PRESELECCIÓN DE CARRIL DE FRENTE Y DERECHA	
Carril 3	1 metro después del inicio de la trayectoria	MARCA VIAL DE PASO DE CEBRA	
Carril 5	1 metro después del inicio de la trayectoria	MARCA VIAL DE PASO DE CEBRA	
Carril 3	1 metro después del inicio de la trayectoria	GRUPO SEMAFÓRICO TIPO 2 (vehículos y peatones)	Mediana, junto a la vía de tranvía, al final del paso de cebra (sentido Carril 3). La carcasa de vehículos orientada en sentido contrario al Carril 3 y la de peatones mirando a la acera opuesta
Carril 5	1 metro después del inicio de la trayectoria	GRUPO SEMAFÓRICO TIPO 1 (vehículos y peatones)	Acera derecha, al final del paso de cebra (sentido Carril 5). Las carcasas de vehículos orientadas en sentido contrario al Carril 5 y la de peatones mirando a la acera opuesta
Vía Acceso	1+ Lp metros antes del final de la trayectoria	MARCA VIAL DE PASO DE CEBRA	

Vía Salida	1 metro después del inicio de la trayectoria	MARCA VIAL DE PASO DE CEBRA	
Vía Acceso	1+ Lp metros antes del final de la trayectoria	GRUPO SEMAFÓRICO TIPO 3 (trnvías)	Mediana, a la izquierda de la vía, antes del paso de cebra. La carcasas orientada en sentido contrario a la Vía de Acceso

- Señalización en ramales secundarios (ambos ramales tienen la misma señalización):

Carril / Vía	Pk de la línea de detención	Señal	Localización del .osg
Carril 2	1+ Lp metros antes del final de la trayectoria	MARCA VIAL DE PASO DE CEBRA	
Carril 2	1+ Lp metros antes del final de la trayectoria	GRUPO SEMAFÓRICO TIPO 1 (vehículos y peatones)	Acera derecha, al final del paso de cebra (sentido Carril 2). Las carcasas de vehículos orientadas en sentido contrario al Carril 2 y la de peatones mirando a la acera opuesta
Carril 2	1+ Lp metros antes del final de la trayectoria	MARCA VIAL DE CEDA EL PASO	1+ Lc antes del paso de cebra
Carril 3	1 metro después del inicio de la trayectoria	MARCA VIAL DE PASO DE CEBRA	
Carril 3	1 metro después del inicio de la trayectoria	GRUPO SEMAFÓRICO TIPO 1 (vehículos y peatones)	Acera derecha, al final del paso de cebra (sentido Carril 3). Las carcasas de vehículos orientadas en sentido contrario al Carril 3 y la de peatones mirando a la acera opuesta

6.INFORMACIÓN PROCESADA DEL ENTORNO.

A continuación se define la información procesada por la Tecnología Modular tanto para un entorno ferroviario como para un entorno urbano.

6.1.GRUPOS DE INFORMACIÓN PARA EL SUBSISTEMA DE SIMULACIÓN FERROVIARIA.

Nodos

Contiene los datos de cada uno de los nodos.

- Identificador del Nodo. Es un entero.

- Tipo de Nodo.
- Nombre del Nodo.

Secciones

Contiene todas las secciones con sus nodos. Está formado por:

- Identificador de Sección. Un entero.
- Nombre de Sección. Es una cadena, por defecto, se formará con los nombres de los nodos izquierda y derecha.
- Identificador de nodo izquierdo.
- Identificador de nodo derecho.
- PK abs de nodo izquierdo.
- PK abs de nodo derecho.
- PK de nodo izquierdo.
- PK de nodo derecho.
- Longitud Visual. Medido sobre la traza de la vía en 3D.
- Número de Vía.
- Número de Posición.

Conexiones

Contiene, para cada nodo, todas las posibles sucesiones de secciones en el sentido de PKs crecientes. Está formado por:

- ID Sección 1
- ID Sección 2
- ID Nodo

Líneas Principales

Contiene:

- ID Línea Principal
- PK,X, Y, Z, alfa, beta, gamma de su punto de inicio.

Gradientes

Contiene el perfil de la vía.

- ID Sección
- PKi abs (PK absoluto inicial)
- PKf abs (PK absoluto final)
- PKi (PK inicial)
- PKf (PK final)
- Longitud
- Gradiente (en milésimas).
- Zi, Zf

Planta

Contiene geometría de la planta de la vía.

- ID Sección
- PKi abs (PK absoluto inicial)
- PKf abs (PK absoluto final)

- PKi (PK inicial)
- PKf (PK final)
- Longitud
- Radio

Puntos de consolidación

- ID Sección.
- ID Punto Consolidación.
- PK
- X, Y, Z

Paralelismo.

- ID Línea
- PK inicial
- PK final
- ID Línea que define el paralelismo
- Distancia de paralelismo

Poligonal.

La poligonal con todos sus segmentos y datos.

- ID Sección
- PK abs inicial
- PK abs final
- PK Inicial
- PK Final
- E
- X, Y, Z, iniciales
- X, Y, Z, finales

Puntos de Posición.

Incluye la información de todos los puntos que sirven para situar elementos sobre la infraestructura de vía y la situación de características de vía (excepto planta y gradientes), instalaciones, etc.

Existen dos tipos de elementos: puntuales y con longitud. Los elementos con longitud se modelan con dos puntos, uno inicial y otro final.

- Identificador Posición. Es un número único para todos los puntos de cualquier sección.
- Identificador de Sección.
- Identificador de extremo. La letra 'I', 'F' ó 'P' dependiendo si es la situación de un extremo inicial, final o de algo puntual.
- Tipo de elemento: por ejemplo, circuito de vía.
- Modo de situación prioritario del Punto. Establece cuál es la información de partida: PK abs, PK, PKR , E, ER .
- PK abs del punto.
- PK del punto.
- E (Espacio) del punto izquierdo.
- Identificador del punto de referencia.

- Identificador de extremo del punto de referencia. La letra 'I', 'F' ó 'P' dependiendo si es la situación de un punto inicial, final o de algo puntual. 'V' (de Vacío) si no se usa.
- Distancia al punto de referencia. Si es positiva el punto situado está a la derecha del punto de referencia y viceversa.
- Lado. Situación del punto respecto de la traza de la vía. Se mira hacia PK crecientes. 'I', 'D', 'M' para izquierda, derecha o en el medio.
- Desplazamiento desde la poligonal. Siempre es positivo. La distancia en metros desde la poligonal en el sentido dado por Lado.
- Sentido. 'T' si el elemento se 've' al recorrer la vía en sentido de PK crecientes. 'F' en caso contrario.
- X, Y, Z, alfa, beta, gamma. En realidad 6 campos para la situación en 3D del punto.

1) Identificador Modelo 3D.

6.2.GRUPOS DE INFORMACIÓN PARA EL SUBSISTEMA DE TRÁFICO.

La información generada para el susbsistema de tráfico consiste en:

- archivos para Tráfico vehicular:
  - Circulacion_carriles.txt: contiene toda la información referente a carriles.
  - Circulacion_intersecciones.txt: contiene toda la información referente a intersecciones.
  - Circulacion_perfiles.txt: contiene toda la información referente a perfiles.
  - Conectividades_tranvia.txt, que también se envía a motor gráfico y contiene toda la información sobre las conectividades en las trayectorias de tranvía.
- archivos para el tráfico de peatones: con la ayuda de la herramienta Malla Editor, desarrollada por la Tecnología Modular para la generación de las áreas no modulares del escenario, se generan:
  - archivo de áreas: contiene toda la información referente a las áreas que afectan a los peatones: estaciones, pasos de cebra..etc
  - archivo de obstáculos: contiene toda la información referente a las áreas que pueden constituir un obstáculo a los peatones con el fin de evitar su colisión.

A continuación se detalla la información de dichos archivos, información que podrá ser editada a través de URBEDIT y/o Malla Editor.

6.2.1.ARCHIVO DE CONFIGURACIÓN DE TRAYECTORIAS: CIRCULACIÓN_CARRILES.TXT.

En este archivo se describe la geometría de las trayectorias que componen el entorno de circulación. Las columnas que componen el archivo son:

NOMBRE:

Dependiendo del tipo de trayectoria que estemos definiendo (un carril, una ruta, una vía de tranvía, una ruta de una rotonda o un carril interior de una rotonda), una trayectoria se nombrará de una forma u otra:

El formato del nombre para trayectorias de los tipos ARCEN, ESTANDAR y MEDIANA es del tipo *CNA1_NI4_2*, donde:

- C es de carril.

- *NA1* es el nodo de inicio del ramal al que pertenece la trayectoria.
- *NI4* es el nodo final del ramal al que pertenece la trayectoria (Nodo Intersección 4).
- *2* es la posición de la trayectoria dentro del ramal. Es par si el sentido de circulación por la trayectoria coincide con el sentido del ramal al que pertenece, siendo 2 el primer número par asignable. Es impar si tiene el sentido contrario al del ramal excepto si es el 1, que puede tener el sentido del ramal o el contrario. En cualquier caso, el sentido de circulación por la trayectoria lo tomamos del campo SENTIDO.

Mientras que el formato para rutas es CNA1_NI4_2_CNA3_NI4_3, donde:

- *CNA1_NI4_2* es el carril de entrada en la ruta.
- *CNA3_NI4_3* es el carril de salida en la ruta.

Las distintas trayectorias que componen una rotonda tienen una nomenclatura ligeramente distinta, pero basada en los mismos principios. Al generar la geometría de la ciudad toda la rotonda está incluida en un mismo nodo, pero de cara al funcionamiento interno del modelo de tráfico será dividida en distintos nodos y ramales interiores de la rotonda. Así pues, una rotonda se compondrá de una serie de nodos interiores (donde se encuentran las entradas y salidas de la misma) y de ramales interiores (tramos de rotonda entre dos nodos interiores). Los nodos en que dividiremos la rotonda llevarán el nombre del ramal que llega a ellos, pero sin la R inicial (es decir, el nombre de la curva constructiva que llega a ese sector de la rotonda). Por ejemplo, si el nodo rotonda se llama NA y uno de los ramales que llegan a él se llama RN1_NA (ramal que une el nodo N1 con el nodo rotonda NA), el nodo interior llevará el nombre de N1_NA. Los carriles de los ramales interiores de la rotonda llevarán nombres del tipo CRN1_NA_NA_N2_2, donde:

- *CR* es de carril de ramal interior de rotonda.
- *N1_NA* es el nodo interior de inicio.
- *NA_N2* es el nodo interior final.
- *2* es el número de carril. Los números de los carriles de ramal interior de rotonda siempre son pares, y el 2 es el más interior.

Las rutas de los nodos interiores de rotonda llevarán un nombre que es la concatenación de los nombre del carril de llegada y del de salida (estos pueden ser carriles de ramales normales o de ramales interiores de la rotonda).

Además de comenzar por los caracteres C o CR, el nombre de una trayectoria puede comenzar por CT. De ser así se trata de una trayectoria de tranvía. Estas son descritas en este mismo fichero y de la misma forma que los carriles de los coches (su nombre es la concatenación de los nodos de inicio y de fin de la trayectoria).

TIPO DE TRAYECTORIA

Indica el tipo de trayectoria que se está describiendo. Los tipos existentes son los siguientes:

- *ARCEN*.
- *ESTANDAR* (carril convencional, exceptuando los carriles de los ramales interiores de las rotondas).
- *CARRIL_ROTONDA* (carriles de los ramales interiores de las rotondas).
- *MEDIANA* (se almacena, pero no se puede circular por ella).
- *RUTA* (exceptuando las rutas de los nodos interiores de las rotondas).
- *RUTA_ROTONDA* (rutas de los nodos interiores de las rotondas).
- *TRANVIA* (vía por la que circulará un tranvía).

TIPO DE TRAMO

Una trayectoria, ya sea carril de una calle o ruta de una intersección, está formada por una serie de tramos consecutivos. Los tramos pueden ser o segmentos rectos o arcos de circunferencia. Cada dos líneas del archivo de circulación_carriles describen un tramo. La primera línea indica el punto de inicio, y en esta columna de dicha línea siempre pondrá *Punto*. La segunda línea indica el punto final y en esta columna se indica si es un tramo segmento o un tramo arco. Por lo tanto, este campo puede tener tres valores:

- *Punto*: es el comienzo del tramo.
- *Recta*: indica el final del tramo y que es un segmento recto.
- *Arco*: indica el final del tramo y que es un arco de circunferencia.

PK

Punto kilométrico: Esta columna sólo hay que rellenarla cuando se trate de trayectorias de tipo ARCEN, ESTANDAR, MEDIANA ó TRANVIA, pero no cuando se trate de trayectorias tipo RUTA, RUTA_ROTONDA, CARRIL_ROTONDA ó RUTA_TRANVIA. Al punto de comienzo del carril se le asigna el Pk 0.0, y a las rectas y arcos se les asigna el Pk final del tramo en cuestión.

COORDENADA X

Coordenada X del punto de comienzo de la trayectoria (si es tipo punto) o el punto final (si es arco o recta).

COORDENADA Y

Coordenada Y del punto de comienzo de la trayectoria (si es tipo punto) o el punto final (si es arco o recta).

COORDENADA Z

Coordenada Z del punto de comienzo de la trayectoria (si es tipo punto) o el punto final (si es arco o recta).

VELOCIDAD MÁXIMA

Velocidad máxima del tramo en m/s.

ANCHURA DE LA TRAYECTORIA

Anchura del carril en metros.

NOMBRE DEL RAMAL/INTERSECCION

Nombre del ramal o intersección de la que forma parte la trayectoria a la que pertenece el tramo: las trayectorias tipo ARCEN, ESTANDAR, MEDIANA ó VIA_TRANVIA pertenecen a ramales, mientras que las trayectorias tipo RUTA, RUTA_ROTONDA, CARRIL_ROTONDA ó RUTA_TRANVIA pertenecen a intersecciones (en el caso de que sea del tipo RUTA_ROTONDA ó CARRIL_ROTONDA, en esta columna se pone el nombre del nodo

rotonda, no el nombre del nodo interior de la rotonda al que pertenezca la ruta o el del ramal interior al que pertenezca el carril). Para ramales el formato es 'RNA1_NI4', donde:

- R indica ramal.
- NA1 es el nodo de inicio.
- NI4 es el nodo final.

Mientras que para intersecciones es 'NI4', donde:

- N indica nodo.
- I4 es el identificador de la intersección.

SENTIDO

Indica si el sentido del carril coincide (1) o no (0) con el del ramal al que pertenece. Si se trata de una ruta, se pone 1.

RADIO DE CURVATURA

Indica el radio de curvatura de los arcos en m (si es tipo arco). En el resto se pone 0.0 (si es punto o recta). Toma valor negativo en arcos dibujados en sentido horario

6.2.2.ARCHIVO DE CONFIGURACIÓN DE CONECTIVIDADES: CIRCULACIÓN_INTERSECCIONES.TXT.

En este archivo se configuran las conectividades entre las trayectorias, es decir, cual es la trayectoria siguiente a otra.

Si se trata de una intersección cada fila configura las conectividades de una ruta, es decir, la trayectoria de llegada a una cierta ruta de una intersección y la trayectoria a la que da a parar dicha ruta.

También puede configurar cómo están conectados carriles de ramales que se siguen uno a otro sin haber una intersección entre medias.

Las columnas que componen el archivo son:

NOMBRE DEL RAMAL/INTERSECCION

Nombre del nodo o del ramal. Normalmente aquí escribiremos el nombre del nodo en el que se encuentra la ruta, con las siguientes excepciones:

- Si la conectividad es de tipo "TRAYECTORIA", este campo no se usa.
- Si la conectividad es de tipo "TRANVIA" y en este campo está el nombre de un ramal, esta fila del archivo configura un carril de tranvía que circula superpuesto a uno o más carriles de tráfico rodado.
- Si la conectividad es de tipo "TRANVIA" y en este campo está el nombre de un nodo, esta fila del archivo configura las conectividades de una ruta de tranvía que se cruza con el tráfico rodado.

TIPO DE CONECTIVIDAD

Se corresponde con el tipo de nodo en el que se encuentra la ruta, a no ser que se trate de conectividad entre dos ramales consecutivos o de vías de tranvía (en ambos casos será conectividad tipo Trayectoria). Los tipos de nodos que hay están basados en la forma en que se genera la geometría de las ciudades. Existen los siguientes tipos de conectividades:

- *Rotonda*
- *Corte*
- *Cruce_T*
- *Corte_mediana*: Reservado para las intersecciones en las que hay una mediana (isleta). MicroTraff buscará trayectorias de tipo MEDIANA en los ramales de acceso. Si las encuentra, las unirá entre sí con otra trayectoria que se considerará una isleta. Todas las rutas que intersecten con la isleta serán eliminadas del entorno de circulación (en rojo en el dibujo).

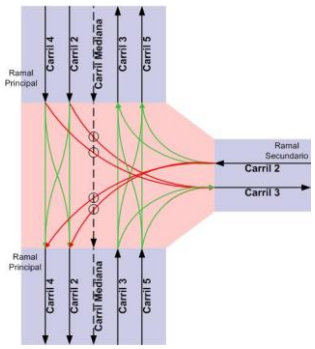


Figura A2.7 Ejemplo de intersección Corte_mediana.

- *Incorporacion*: Reservado para las intersecciones de 3 ramales en las que el secundario sólo tiene carriles de entrada a la intersección. En nodos de este tipo MicroTraff actuará como si hubiera una isleta uniendo las lineas medias de los dos ramales principales, es decir, todas las rutas que crucen el centro de la intersección serán eliminadas (en rojo en el dibujo).

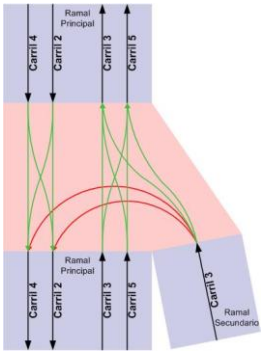


Figura A2.8 Ejemplo de Incorporación.

- *Salida*: Reservado para las intersecciones de 3 ramales en las que el secundario sólo tiene carriles de salida de la intersección. En nodos de este tipo MicroTraff actuará como si hubiera una isleta uniendo las lineas medias de los dos ramales principales, es decir, todas las rutas que crucen el centro de la intersección serán eliminadas (en rojo en el dibujo).

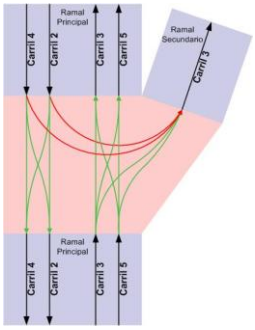


Figura A2.9 Ejemplo de Salida.

- *Aumento_carriles*
- *Reduccion_carriles*
- *Trayectoria* (para conectar dos ramales consecutivos y las vías de los tranvías).
- *Tranvía* (para configurar los carriles y las rutas de los tranvías).

NOMBRE DEL CARRIL DE LLEGADA

Nombre del carril de llegada a la ruta. Si es una conectividad tipo Trayectoria, aquí escribiremos el nombre del carril anterior al carril que figure en la siguiente columna. Hay que tener en cuenta que no es lo mismo que el carril tenga el sentido del ramal o el contrario.

NOMBRE DE LA RUTA

Rutas que se pueden tomar si llegamos a la intersección por la trayectoria de la columna anterior y salimos por la trayectoria indicada en la columna siguiente. Si la conectividad es de tipo Trayectoria no lo hacemos, pues no pertenece a ninguna intersección.

NOMBRE DEL CARRIL DE SALIDA

Carril de salida de la ruta: Si es una conectividad tipo Trayectoria, aquí escribiremos el nombre del carril siguiente al carril que figure en la anterior columna..

6.2.3.ARCHIVO DE CONFIGURACIÓN DE PERFILES: CIRCULACIÓN_PERFILES.TXT.

Este archivo contiene las cotas z de las diferentes trayectorias, es decir, sus perfiles.

Sus columnas son:

- Nombre del ramal, de la ruta, de la trayectoria de tranvía, de la ruta interior de rotonda o del carril interior de rotonda.
- Tipo de trayectoria. Puede ser *RAMAL*, *RUTA*, *TRANVIA*, *RUTA_ROTONDA*, *CARRIL_ROTONDA*.
- Pk: punto kilométrico en el que se da la cota Z.
- Z: cota Z en ese punto.

6.2.4.ARCHIVO DE CONFIGURACIÓN DE CONECTIVIDADES ENTRE LAS VÍAS DE TRANVÍA: CONECTIVIDADES_TRANVIA.TXT.

Este archivo está estructurado igual que el archivo de conectividades entre las trayectorias del tráfico rodado. En un cambio de aguja una trayectoria de tranvía termina y comienzan otras. Por cada cambio de aguja existente en el entramado de vías de tranvía hay entonces una serie de conectividades. En este archivo tendremos una línea por cada conectividad entre vías de tranvía que haya en el entorno.:

- La primera columna será el nombre del nodo o cambio de aguja, que es intrascendente para MicroTraff.
- En la segunda columna se pone el nombre de la trayectoria de entrada al nodo.
- En la tercera columna se pone el nombre de una posible trayectoria de salida del nodo.

6.2.5.ARCHIVO DE ÁREAS DE PEATÓN.

Para poder procesar el movimiento de los peatones Microtraff necesita delimitar unas áreas tridimensionales. El contorno de estas áreas y otras informaciones adicionales, separadas mediante punto y coma, son las que vienen en los siguientes ejemplos:

LNPC12T_NML136MARIA_7_ANDEN_DER_111;2;i;0;87,40,-96;97,52,-96;10,80,-97;21,67,-97.6; ***E;***87,40,-96;97,52,-96;110,77,-97.1256;124,47,-97;

GA_RNTI22_NTI23_POS_4-2_PASO_PEA_8;1;e;0;220.7599972,2950.195756,-109.6928621;217.0674609,2953.566984,-109.692872;207.3583251,2942.932479,-109.692872;211.0508613,2939.561252,-109.6928621;

Las columnas que contiene este archivo, separadas mediante punto y coma, son:
• Identificativo de área: cadena que identifica el área de forma única.

• Tipo de área:

1. **Acera**. Los peatones de este área no interactúan con el tráfico.
2. **Paso de cebra**. Atraviesan un ramal, por lo que pueden interactuar con el tráfico que circule por dicho ramal.
3. **Parada**. Los peatones se amontonan en esta área a la espera de que un conductor llegue y abra las puertas. El área dada se expande en dirección a la carretera para ocupar parte de la misma, a fin de que las puertas del vehículo que pare en ella coincidan con alguna celda de carretera. Esta área de expansión se describe, tras insertar el encabezado **E;** dando los puntos de la poligonal del contorno.
4. **Ilegal**. Área utilizada en la incidencia del "*peatón suicida"* (peatón que cruza por donde no debe). Son áreas que cruzan la carretera o la vía por un lugar no señalizado, y que sólo puede utilizar el peatón marcado como suicida. Necesitan ser adyacentes a alguna acera, para tener una fuente de peatones de la que elegir al suicida.
5. **Descansillo**. Son partes de acera situados entre pasos de cebra, que utilizan los peatones para cruzar. Se diferencian de las aceras en el rutado (aquí no pueden deambular y deben salir por el lado contrario al que entraron, como en los pasos de cebra), y se diferencian de los pasos de cebra en que aquí sí que pueden esperar en el borde a que no pasen coches (como en las aceras).

6. **Interior/exterior**: sirve para aplicar efectos (como lluvia, nieve, iluminación…) según si el área está en el interior de algún edificio o en la calle.

• Dirección del transporte: indica por dónde viene el transporte para que los peatones miren hacia ahí. Aunque deba estar, sólo se usa su valor en las paradas, por lo que sólo ahí hay que calcularlo (el resto se pone 0 por defecto).

- 0: Viene por el vértice del área de menor valor en X (abcisas). Si hay varios vértices que tengan este valor, entonces se coge el de mayor valor en Y (ordenadas).
- 1: Viene por el vértice del área de mayor valor en X (abcisas). Si hay varios vértices que tengan este valor, entonces se coge el de menor valor en Y (ordenadas).

• N coordenadas de vértices: siendo el área un polígono cerrado formado por N vértices, se ponen los vértices en orden separados por punto y coma. Cada vértice consta de tres números separados mediante una coma, que representan la coordenada 'x', la coordenada 'y' y la 'z' de dicho vértice.

En el caso de paradas, en primer lugar vendrá dado el conjunto de puntos que forman el andén, y después, separados por **E,** para indicar expansión, vendrán los puntos que forman la expansión. No importa cuál sea el primer punto por el que se empiece a describir el polígono, ni si se describe girando a derechas o a izquierdas, lo único importante es que los puntos vayan seguidos en orden como si se dibujase a partir de ese primer punto. Con la expansión ocurre igual, pero además es importante que sea adyacente al andén (si hay hueco entre las dos áreas, puede que al hacer la rejilla no haya conexión, por lo que los peatones no podrían pasar a ese área).

6.2.6.ARCHIVO DE OBSTÁCULOS DE PEATÓN.

Cuando se crea un área, puede contener obstáculos en su interior (expendedoras, paredes, bancos...). Este archivo contiene todos los obstáculos que hay en el entorno, y las columnas que contiene, separadas mediante punto y coma, son:

9;ESAAN_D_99;-39.7,3167.8,-10;-39.1,3168.4,-10;-42.7,3172,-10;-43.3,3171.4,-10;

• Identificativo de obstáculo: número que identifica el obstáculo de forma única.
• Área asociada: identificativo del área en la que se encuentra el obstáculo.
• N coordenadas de vértices: siendo el obstáculo un polígono cerrado formado por N vértices, se ponen los vértices en orden separados por punto y coma. Cada vértice consta de tres números separados mediante una coma, que representan la coordenada 'x', la coordenada 'y' y la 'z' de dicho vértice (la 'z' debe aparecer, aunque como no se utiliza, puede tener cualquier valor).

El obstáculo no tiene porqué estar contenido totalmente dentro del área, sino que puede sobresalir de ella.

6.3.GRUPOS DE INFORMACIÓN PARA EL SUBSISTEMA DE MOTOR GRÁFICO.

La información generada para el motor gráfico consistirá en:

- Data.dat: archivo de texto plano con el grafo de la escena.

- Archivos con la información de trayectorias y conectividades. Ambos archivos tendrán un aspecto similar al grupo de información Poligonal y Conexiones respectivamente, vistos en el apartado 6.1.

6.4.GRUPOS DE INFORMACIÓN PARA EL SUBSISTEMA INFOGRAFISTA.

La información generada para el grupo infografista consistirá en:

- Señalización a modelar.
- Definición del entorno: será un archivo con un contenido similar al archivo visto para el subsistema de simulación denominado Puntos de Posición (apartado 6.1), en el cuál el elemento a posicionar será un elemento de entorno.

7.TECNOLOGÍA PARA PERSISTENCIA.

El formato de los archivos comunes para configuración del escenario será ASCII. Los archivos de texto constarán de tantas líneas como elementos del grupo de datos y de tantas columnas como datos se indican en la definición de los grupos. Las columnas se separarán por tabuladores.

Los archivos se leerán en cualquier tipo de aplicación de edición, según la finalidad, entre ellas: Excel, Access, COCO, aplicaciones en VB ó VC ++. Es requisito que la edición de los archivos mantenga la coherencia de la información contenida y que la información se guarde en los archivos fuentes originales. Dependiendo de las posibilidades del formato se pondrán comentarios. Cada uno de los archivos estará bajo control de configuración.

Apéndice Grupos Semafóricos

Grupo Semafórico Tipo 1:

Soporte a suelo en L con 3 carcasas: 2 de ellas con, cada una, una columna de tres focos viarios para los vehículos (rojo, amarillo, verde) y una carcasa con una columna de dos focos para los peatones (rojo, verde).

Grupo Semafórico Tipo 2:

Soporte a suelo con 2 carcasas: una con una columna de tres focos viarios para los vehículos (rojo, amarillo, verde) y la otra con una columna de dos focos para los peatones (rojo, verde).

Grupo Semafórico Tipo 3:

Soporte a suelo con 1 carcasa: una columna de tres focos viarios para los tranvías (triángulo, raya vertical, raye horizontal).

ANEXO 3. GENFAM: GENERADOR DE FAMILIAS.

A continuación se detallan los distintos diálogos que componen la herramienta GENFAM, el Generador de Familias desarrollado por la Tecnología Modular para CITEF.

1.DIÁLOGO DE EDICIÓN DE FAMILIAS.

Una familia se compone de una serie de macrosolevados situados en distintos entornos de proximidad y lateralidad (E1 izq, E3 derecha, E0 centrado, etc) . Cada macrosolevado está formado por un conjunto de microsolevados. Estos microsolevados se forman por la extrusión de una serie de perfiles transversales a lo largo de una serie de recorridos. A través del diálogo de Edición de Familias se controlan estos parámetros. Para acceder a este diálogo se selecciona una familia de la base de datos y se pulsa el botón Editar Familia.

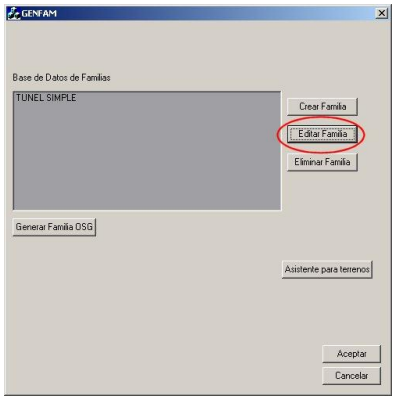


Figura A3.1 Generador de familias.

En el diálogo de Edición de familias, las zonas señaladas en rojo vendrán definidas por la base de datos de familias y serán invariables desde este editor. Las zonas verdes serán las que ofrezcan la posibilidad de ser editadas.

Recorrido elegido:

De la base de datos de recorridos, marcado en azul, se eligen aquéllos sobre los que se va a realizar la extrusión. Tras seleccionar un recorrido de la lista se pulsa el botón señalado con el círculo rojo, para añadirlo. Se pueden añadir tantos recorridos cómo se deseen. El Editor de Recorridos, que se verá posteriormente, permite crear y editar recorridos ya existentes.

Macrosolevados elegidos:

Para cada uno de los entornos de proximidad (E0, E1,…) y cada uno de los lados (Izquierda, Derecha, Centrado) se elige de la base de datos de macrosolevados, marcado en azul, aquellos que se desean extrusionar. Tras seleccionar un macrosolevado de la lista se pulsa el botón señalado con el círculo rojo. El Editor de Macrosolevados, que se verá posteriormente, permite crear y editar macrosolevados ya existentes.

Angulo máximo de suavizado:

Determina el ángulo máximo que pueden formar dos normales de triángulos contiguos, permitiendo así controlar, el número de polígonos del módulo.

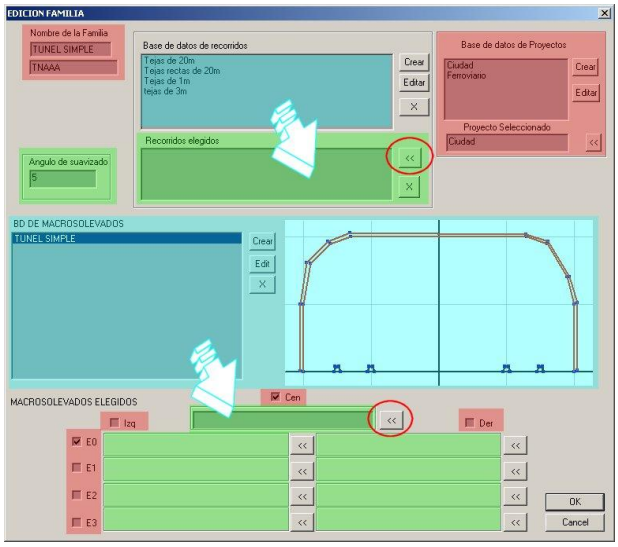


Figura A3.2 Editor de familias.

2.DIÁLOGO DE EDICIÓN RECORRIDOS.

Los recorridos a lo largo de los cuáles van a ser extrusionados los perfiles transversales pueden ser editados a través de este diálogo. Los parámetros editables serán la longitud de la cuerda y el ángulo de suavizado.

Figura A3.3 Editor de recorridos.

3.DIÁLOGO DE EDICIÓN DE MACROSOLEVADOS.

Es posible añadir nuevos microsolevados a un macrosolevado. Para ello se selecciona éste de la base de datos, marcada en azul, y se pulsa el botón señalado con el círculo rojo. El microsolevado añadido se inserta por defecto en el origen de coordenadas. Se puede desplazar del origen hasta la posición deseada utilizando los controles existentes en la parte inferior del diálogo. Es posible crear y editar microsolevados existentes a partir del editor de microsolevados.

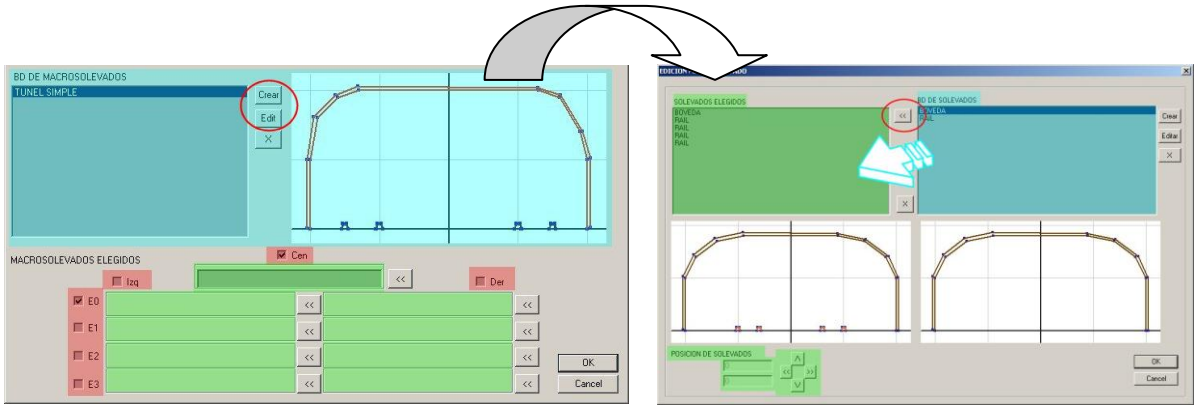


Figura A3.4 Editor de macrosolevados.

4.DIÁLOGO DE EDICIÓN DE MICROSOLEVADOS.

Un microsolevado consiste en la extrusión de unos perfiles transversales a lo largo de un recorrido. Cada microsolevado tiene asociado una textura. Para editar los parámetros que definen el microsolevado existe el cuadro de Edición de Microsolevados, al que se accede a través de los botones marcados con un círculo rojo.

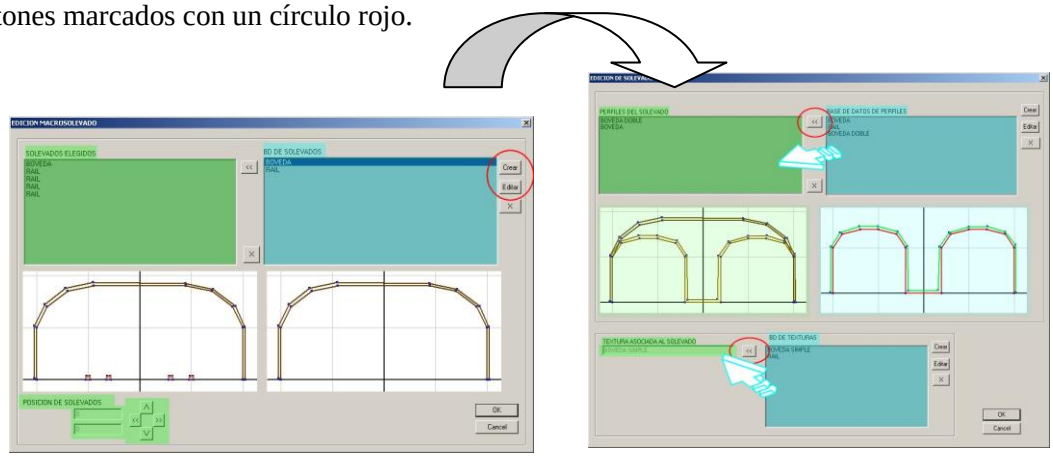


Figura A3.5 Editor de microsolevados.

Perfiles del microsolevado:

En este diálogo podemos configurar la lista de perfiles del microsolevado añadiéndolos desde la base de datos, marcada en azul, pinchando sobre el perfil elegido y pulsando en botón marcado con el círculo rojo. Si alguno de los perfiles se quisiera editar o crear uno nuevo se puede hacer a través del diálogo de Edición de Perfiles.

Textura asociada al solevado:

A través de este diálogo se introduce la textura que se aplica al microsolevado, seleccionándola de la base de datos. Si alguna de las texturas se quisiera editar o crear una nueva se puede a través del dialogo de Edición de Texturas.

5.DIÁLOGO DE EDICIÓN DE TEXTURAS.

Una textura se compone de un archivo de imagen y unos parámetros de tiling y visualización. A través del dialogo de Edición de texturas se controlan estos parámetros. Para acceder a él se pulsa sobre los botones de Crear y Editar marcados con el círculo rojo. Las zonas marcadas de verde son las editables por el usuario.

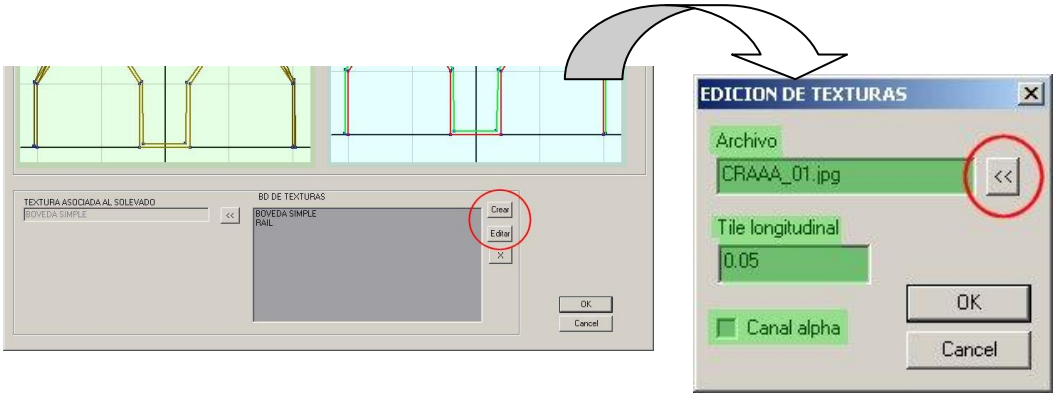


Figura A3.6 Editor de texturas.

Archivo:

Es un archivo de imagen elegido por el usuario. La imagen debe ser tal que el eje horizontal de la imagen coincida con la dirección de extrusión.

Tile longitudinal:

Determina el número de veces por unidad de longitud que se repite la textura en la dirección de la extrusión.

Canal Alpha:

Si activamos esta casilla el filtro de mipmapping se coloca en linear para evitar efectos de pixelazo en el recorte alpha de la textura.

6.DIÁLOGO DE EDICIÓN DE PERFILES.

Cada perfil comprende dos poligonales: una poligonal principal, que se extruye generando el microsolevado y una poligonal auxiliar que se emplea para generar la solapa. En el diálogo de la Edición de Perfiles se pueden definir ambas poligonales. Para acceder al diálogo se pulsa el botón marcado con el círculo rojo. Las zonas marcadas de verde son las editables por el usuario.

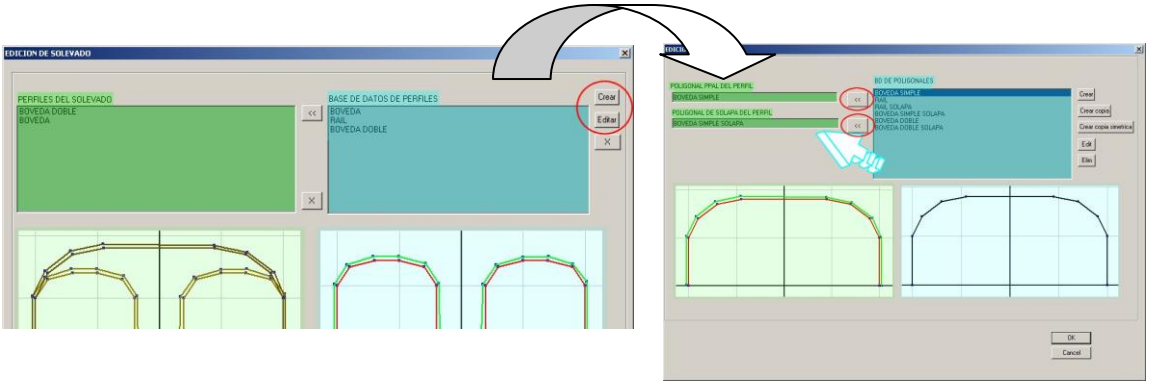


Figura A3.7 Editor de perfiles.

Poligonal principal y de solapa:

Se eligen de la base de datos, marcada en azul, las poligonales que se desean añadir al perfil a través de los botones marcados con el círculo rojo. Si se quiere editar alguna de las poligonales o crear una nueva se puede hacer a través del diálogo de Edición de poligonales.

7.DIÁLOGO DE EDICIÓN DE POLIGONALES.

La poligonal es un conjunto ordenado de puntos. Cada uno de los puntos tendrá información geométrica en el plano transversal (y, z) así como una tercera coordenada (m) que hace referencia a la coordenada transversal de textura. El cuadro de Edición de poligonales se encarga de facilitar la entrada de estos puntos, tanto de sus coordenadas geométricas como de textura. Para acceder al diálogo se pulsa sobre los botones de Crear/Editar marcados con el círculo rojo.

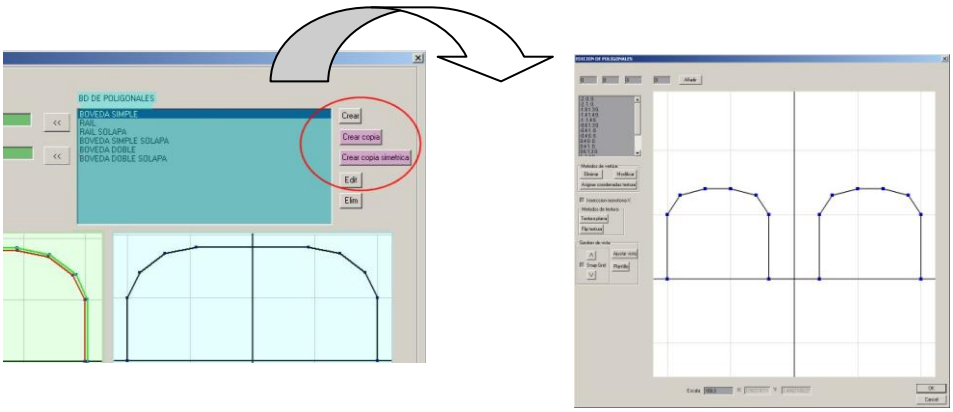


Figura A3.8 Editor de poligonales.

Mediante la opción “Crear Copia” el diálogo de edición de poligonales crea una copia de la poligonal elegida y mediante “Crear copia simétrica” crea una copia simétrica respecto del eje y de la poligonal elegida. En cualquiera de los casos de abre un diálogo como el que se muestra en la figura anterior.

Sobre la zona de la cuadrícula podemos definir la poligonal con las siguientes funcionalidades:

- Doble clic botón izquierdo: Introducción de un punto de la poligonal. El orden de introducción en la lista puede ser de dos formas:

- Si la inserción monótona está activada el punto se inserta por orden creciente de la coordenada y.
- Si esta desactivada se inserta después del último punto insertado.

- Pinchar y Arrastrar botón izquierdo sobre un vértice: se desplaza el vértice de la poligonal.
- Pinchar y arrastrar botón derecho: se desplaza el encuadre.
- Desplazamiento de la rueda de scroll: Zoom.
- Hacer clic sobre un vértice: Seleccionar un vértice. Una vez seleccionado se puede:
  - Eliminar: pulsando el botón "Eliminar".
  - Modificar sus parámetros de coordenada espacial y de textura: pulsando el botón "Editar", se abre el cuadro de dialogo siguiente.

Figura A3.9 Editor de vértices.

- Asignar la coordenada de textura de forma visual. Al pulsar el botón "Asignar coordenada" aparece el siguiente cuadro de diálogo que muestra la textura asociada al solevado.

Figura A3.10 Editor de texturas.

Haciendo doble clic sobre cualquier punto de la textura se asigna automáticamente su coordenada de textura (u,v) al punto seleccionado.

Existe un grupo de comandos dedicados a herramientas de visualización

- "Plantilla": Se abre un cuadro de diálogo en el que elegimos una poligonal de la base de datos que nos sirve como plantilla.

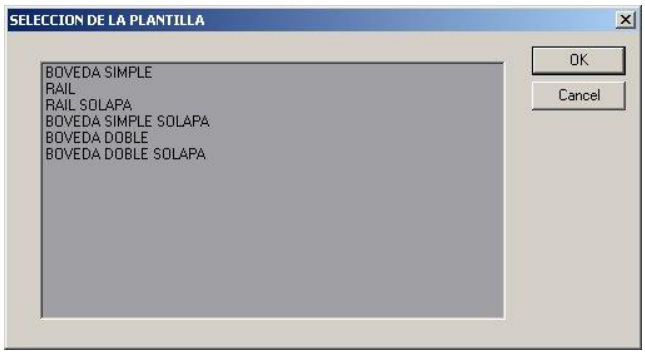


Figura A3.11 Selección de plantillas.

- "Ajustar Vista": Calcula el zoom y el encuadre para que quede ajustado a la poligonal que se esta editando y a la plantilla.
- "Snap Grid": Activa la configuración de selección en la que los desplazamientos solo pueden darse sobre los nodos de la cuadricula.
    - Se puede aumentar o reducir el refinamiento de la cuadrícula manualmente con los botones situados encima y debajo. Nótese que la cuadricula posee un algoritmo de transformación exponencial que ajusta la densidad de esta al zoom de forma dinámica.

Existe un grupo de comandos dedicados herramientas de mapeado**.**

- "Textura plana": Asigna coordenadas de textura a los vértices del perfil proyectando verticalmente la textura ajustando su tamaño al perfil.
- "Flip textura": Varia las coordenadas de texturas para mostrar la textura en espejo.

8.ARCHIVO DE CONFIGURACIÓN DE FAMILIAS.

Una vez generada la familia, GENFAM rellena el archivo de configuración cfg_TipoSubfamilias.txt, el cual almacena toda la información que la Tecnología Modular necesita para generar el entorno. Los campos del mismo son los siguientes:

- m_nombre_familia: define un nombre para la familia. (TUNELDOBLE, TUNEL SIMPLE, ESTACIONSIMPLE, TERRENO..)
- m_id_familia: dos siglas que se emplean para identificar la familia en la nomenclatura del módulo (TD,TD,ES,TR..)
- m_nombre_subfamilia: define un nombre para la subfamilia. (TUNEL_BOVEDA, TUNEL_ELIPSE..)
- m_id_subfamilia: tres siglas que se emplean para identificar la familia en la nomenclatura del módulo.
- m_num_vias_abarcadas: indica el número de vías sobre las que asienta el módulo. Por ejemplo, en el caso de un túnel doble será dos.
- m_E0_afectado: indica si la familia ocupa el nivel de proximidad 0 (sobre la vía).
- m_E1_afectado: indica si la familia ocupa el nivel de proximidad 1.
- m_E2_afectado: indica si la familia ocupa el nivel de proximidad 2.
- m_E3_afectado: indica si la familia ocupa el nivel de proximidad 3.
- m_lado: indica el lado de la vía sobre el que se coloca la familia: izquierda, derecha, centrado.
- m_texturas: listado de texturas posibles a emplear.
- m_id_texturas: listado con identificadores de texturas posibles a emplear: dos siglas para identificar cada textura en la nomenclatura del módulo.
- m_listado_taludes: listado de taludes que definen la familia.
- m_id_proyecto: dos siglas que se emplean para identificar el proyecto al que pertenece la familia en la nomenclatura del módulo.
- m_block: indica si la familia es un elemento de tipo bloque, es decir, no modular. Ejemplo: las estaciones de una línea de Metro de Madrid .
- m_puntual: indica si la familia es un elemento puntual, es decir, su longitud no es relevante . Ejemplo: la familia Arbol.
- m_longX: dimensión de la familia en X.
- m_longY: dimensión de la familia en Y.
- m_longZ: dimensión de la familia en Z.
- m_exterior: define si la familia es de interior, exterior o de ambos tipos.
- m_iluminacion_estacion: indica si la familia ha de llevar la iluminación especial que llevan las estaciones.

ANEXO 4. MATEMÁTICA DE LOS SHADER 3D.

La nomenclatura que aquí se expone es la que se usará en todo el tratamiento de shaders.

1.SISTEMAS DE COORDENADAS.

El primer concepto es de espacio de coordenadas global o absoluto que se refiere al sistema raíz del entorno virtual, equiparable a un sistema inercial. En ese entorno vamos a definir una terna a derechas de vectores unitarios ortogonales (i, j, k), cada uno en la dirección de los ejes x, y, z respectivamente.

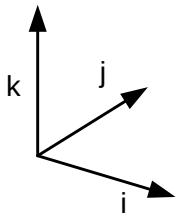


Figura A4.1 Triedro referencia.

Los objetos tienen cada uno un sistema de referencia interno denominado local. Posteriormente se comentará cómo pasar de un sistema de referencia global a otro local.

2.PUNTOS Y COORDENADAS.

Un punto del espacio P se puede identificar por sus coordenadas con respecto a un triedro de referencia (i, j, k):

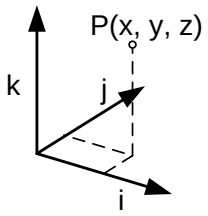


Figura A4.2 Coordenadas de un punto.

3.NUMERACIÓN DE PUNTOS Y REFERENCIAS.

En este desarrollo se van a utilizar distintos puntos y distintos sistemas de referencia. Los diferentes puntos se identificarán por un subíndice, así se tiene que:

$$\begin{matrix} P_1 \equiv (x_1, y_1, z_1) \\ P_2 \equiv (x_2, y_2, z_2\ ) \\ \vdots \\ P_n \equiv (x_n, y_n, z_n\ ) \end{matrix} .(1)$$

Representan **n** puntos distintos del espacio. Cada uno de estos puntos puede representarse en **m** referencias distintas:

$$\begin{matrix} (i_1, j_1, k_1) \\ (i_2, j_2, k_2\ ) \\ \vdots \\ (i_m, j_m, k_m\ ) \end{matrix} .(2)$$

Para generalizar la nomenclatura un punto se identificará con un superíndice el sistema respecto al cual están expresadas sus coordenadas:

$$P_n^m (x_n^m, y_n^m, z_n^m) .(3)$$

Lo anterior es la expresión del punto **n**, $\mathbf{P_n}$, en el sistema de coordenadas m.

4.CAMBIOS DE BASE.

Se va a ver ahora como se cambian las referencias mediante giros ordenados alrededor de sus correspondientes ejes o vectores unitarios. En principio se va a partir de un punto $\mathbf{P_1}$ referido a un sistema inicial (i_1, j_1, k_1):

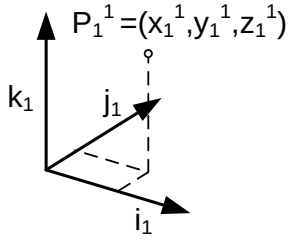


Figura A4.3 Situación inicial antes de cambio de base.

4.1.GIRO DE ORIENTACIÓN γ.

El primer cambio de base consiste en un giro **γ** alrededor del eje $\mathbf{k_1}$:

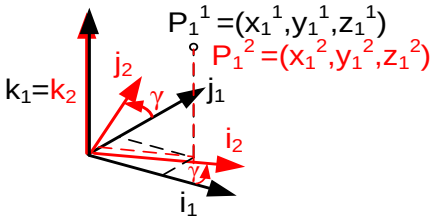


Figura A4.4 Cambio de orientación.

Se obtiene un nuevo sistema de referencia (**i₂, j₂, k₂**) cuya expresión es la siguiente:

$$i_2^1 \equiv (\cos\gamma, sen\gamma, 0)$$
$$j_2^1 \equiv (-sen\gamma, \cos\gamma, 0) \quad .(4)$$
$$k_2^1 \equiv (0,0,1)$$

Del algebra matricial se sabe que la expresión de una base en coordenadas de otra forma la matriz de cambio de base de una a otra $\mathbf{M^{21}}$, que es la matriz para obtener las coordenadas de un punto en el sistema 2 a partir de las coordenadas del sistema 1:

$$M^{21} \equiv \begin{bmatrix} \cos\gamma & sen\gamma & 0 \\ -sen\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$P_1^2(x_1^2, y_1^2, z_1^2) = M^{21} \cdot P_1^{1T} \begin{pmatrix} x_1^1 \\ y_1^1 \\ z_1^1 \end{pmatrix} \quad .(5)$$

Hay que recordar que el punto no varía y que las coordenadas traspuestas identifican igualmente al punto.

4.2.GIRO DE PENDIENTE β.

El segundo cambio de base consiste en un giro **β** alrededor del eje **j₂**:

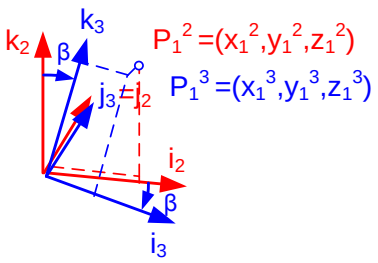


Figura A4.5 Cambio de pendiente.

Ahora se obtiene la expresión de la matriz de cambio de base $\mathbf{M^{32}}$, que es la matriz para obtener las coordenadas de un punto en el sistema 3 a partir de las coordenadas del sistema 2:

$$M^{32} \equiv \begin{bmatrix} \cos\beta & 0 & -sen\beta \\ 0 & 1 & 0 \\ sen\beta & 0 & \cos\beta \end{bmatrix} \quad (6)$$

$$P_1^3(x_1^3, y_1^3, z_1^3) = M^{32} \cdot P_1^{2T} \begin{pmatrix} x_1^2 \\ y_1^2 \\ z_1^2 \end{pmatrix}$$

4.3.GIRO DE PERALTE α.

El tercer cambio de base consiste en un giro **α** alrededor del eje $\mathbf{i_3}$:

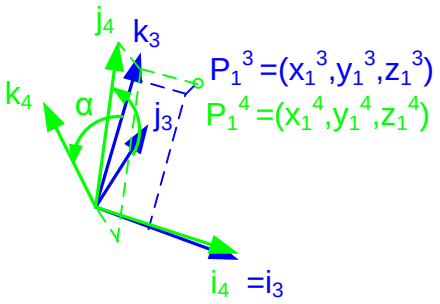


Figura A4.6 Cambio de peralte.

La expresión de $\mathbf{M^{43}}$ es:

$$M^{43} \equiv \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & sen\alpha \\ 0 & -sen\alpha & \cos\alpha \end{bmatrix} \quad (7)$$

$$P_1^4(x_1^4, y_1^4, z_1^4) = M^{43} \cdot P_1^{3T} \begin{pmatrix} x_1^3 \\ y_1^3 \\ z_1^3 \end{pmatrix}$$

Se debe recordar que las matrices $\mathbf{M^{ij}}$ son ortogonales, es decir que su inversa es su transpuesta $\mathbf{M^{ij} = M^{jiT}}$. En este caso se entiende fácilmente porque las matrices inversas deshacen el giro de un sistema a otro, lo que equivale a realizar el giro sobre el mismo eje pero en sentido contrario, β a –β, por ejemplo. La matriz que es antisimétrica y la función seno impar, al cambiar el ángulo se obtiene la transpuesta como establece obviamente la teoría algebraica básica.

5.GIRO DE UN PUNTO.

Lo que interesa ahora es ver como se obtienen las coordenadas de un punto cuando gira alrededor de un eje y permanece inmóvil el sistema de referencia.

5.1.GIRO DE ORIENTACIÓN γ.

El primer cambio consiste en un giro **γ** alrededor del eje $\mathbf{k_1}$:

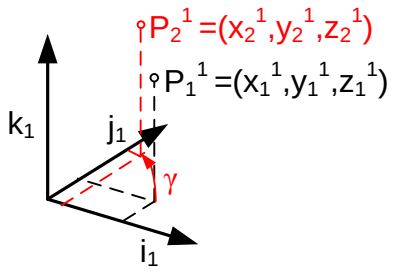


Figura A4.7 Giro 1 de un punto.

Hay que notar que las coordenadas del punto girado $\mathbf{P_2^1\ (x_2^1, y_2^1, z_2^1)}$ son las mismas que si se girara el sistema (i_1, j_1, k_1) un ángulo – **γ** dejando el punto quieto. Si se llama $\mathbf{M_\gamma}$ a la matriz que permite obtener $\mathbf{P_2^1\ (x_2^1, y_2^1, z_2^1)}$, es inmediato deducir que es la misma que $\mathbf{M^{12}}$, la matriz para pasar del sistema 2 al 1:

$$M_\gamma = M^{21T} = M^{12} \equiv \begin{bmatrix} \cos\gamma & -sen\gamma & 0 \\ sen\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

$$P_2^1(x_2^1, y_2^1, z_2^1) = M_\gamma \cdot P_1^{1T} \begin{pmatrix} x_1^1 \\ y_1^1 \\ z_1^1 \end{pmatrix}$$

5.2.GIRO DE PENDIENTE β.

El segundo movimiento consiste en un giro **β** alrededor del eje $\mathbf{j_1}$. Siguiendo razonamientos análogos al apartado anterior se obtiene $\mathbf{M_\beta}$:

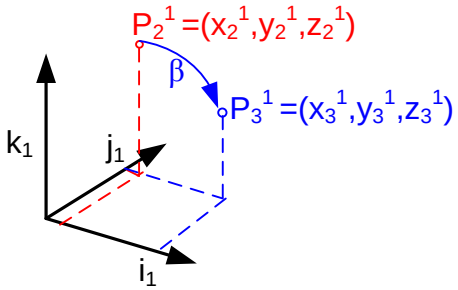


Figura A4.8 Giro 2 de un punto.

$$M_\beta = M^{32T} = M^{23} \equiv \begin{bmatrix} \cos\beta & 0 & sen\beta \\ 0 & 1 & 0 \\ -sen\beta & 0 & \cos\beta \end{bmatrix} \quad (9)$$

$$P_3^1(x_3^1, y_3^1, z_3^1) = M_\beta \cdot P_2^{1T} \begin{pmatrix} x_2^1 \\ y_2^1 \\ z_2^1 \end{pmatrix}$$

5.3.GIRO DE PERALTE α.

El tercer movimiento consiste en un giro **α** alrededor del eje $\mathbf{i}_1$. Siguiendo razonamientos análogos al apartado anterior se obtiene $\mathbf{M}_\alpha$:

$$M_\alpha = M^{43T} = M^{34} \equiv \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -sen\alpha \\ 0 & sen\alpha & \cos\alpha \end{bmatrix} \quad (10)$$

$$P_4^1(x_4^1, y_4^1, z_4^1) = M_\alpha \cdot P_3^{1T} \begin{pmatrix} x_3^1 \\ y_3^1 \\ z_3^1 \end{pmatrix}$$

ANEXO 5. LENGUAJE DE MACROS.

1. **macro_carga_dat**

```
CargaDat -URL_archivo_.DAT-
```

Permite la carga anidada de un archivo .DAT
La ***URL del archivo*** es su nombre y localización dentro del sistema de archivos. Esta macro permite tener múltiples archivos dats separados y cargarlos anidadamente desde uno principal.

2. **macro_efectos**

```
Efecto _nombre_del_efecto { opcional }
```

Da de alta el efecto especificado en ***_nombre_del_efecto.***

3. **macro_fx_cielo**

```
FXCielo cielo {bool}
```

La creación de un entorno ambiental con cielo, nubes y distinto horario es creada automáticamente por elmotor gráfico. Esta macro permite ajustar las siguientes opciones:
Si bool = 0, entonces se renderizara el cielo en un cubemap.
Si bool = 1, se representa el cielo usando un shader en una esfera.
Si bool = 2, entonces se hara cálculo de luces pero no se dibujara el cielo.

4. **macro_group**

Especifica un nodo de agrupación.

```
group <name> {n,o1,o2,...,on,parent(x,y,z,rx,ry,rz)}
group <name> {n,o1,o2,...,on,parent(euler,x,y,z,rx,ry,rz)}
group <name> {n,o1,o2,...,on,parent(quaternio,x,y,z,rx,ry,rz)}
```

Donde:

<name> ,es el nombre del grupo.
N , número de objetos en la agrupacion.
Oi , file(URL_file)/instanceof(object). Tantos como sea el valor de n.
parent ,nodo del cual este grupo es hijo.
X,y,z,rx,ry,rz ,posición y rotación del objeto.

5. **macro_lod**

Los objetos LOD permiten la creación de objetos con múltiples niveles de detalle. A cada nodo lod se le asocian varias geometrías que representan visualmente al objeto asociado a cada nivel. La permutación entre los niveles de detalle es realizada automáticamente por el motor gráfico basándose en la distancia del punto de vista al objeto, según las distancias definidas en la declaración del lod. Su sintaxis es la siguiente:

```
LOD <name> {n,o1,d1,o2,d2,...,on,dn,parent(x,y,z,rx,ry,rz)}
```

```
LOD <name> {n,o1,d1,o2,d2,...,on,dn,parent(euler,x,y,z,rx,ry,rz)}
LOD <name> {n,o1,d1,o2,d2,...,on,dn,parent(quaternio,x,y,z,rx,ry,rz)}
```

Donde:
n , número de niveles de detalle,
o_i , file(..) ó instanceof(<name>)
d1,d2,...,dn , son las distancias a las cuales se permuta de una representación a otra de mayor detalle (mayor cercanía al objeto)**.**

6. macro_lodPK

La macro LodPK permite la activación/desactivación de un nodo a partir de las distancias del Punto Kilométrico.

```
LodPK nombre{ini1,fin1,ini2,fin2,nodo_que_cuelga}
```

Donde:
nombre , es el nombre de una objeto/grupo ya definido.
ini1/fin1 ,PK de inicio y fin de la trayectoria1 durante el cual el nodo se representa.
ini1/fin2 ,PK de inicio y fin de la trayectoria2 durante el cual el nodo se representa.
nodo_que_cuelga ,es el nodo del que cuelga este LodPK.

7. macro_object

Los objetos de la escena se definen por medio del comando object. Las geometrías se pueden cargar desde un fichero o hacer copias (instancias) de objetos previamente creados. La sintáxis del comando es la siguiente:

```
object <name> {obj,parent(x,y,z,rx,ry,rz)}
object <name> {obj,parent(euler,x,y,z,rx,ry,rz)}
object <name> {obj,parent(quaternio,x,y,z,rx,ry,rz,rw)}
```

Siendo;
obj = file(<nombre de archivo>,factor_escala,Tipo)
obj = instanceof(<nombre de un objeto ya creado>,factor_escala,Tipo)

Los formatos aceptados en file son los aceptados por OSG, y sus plugins. Las copias (instancias) de otros objetos son útiles para la construcción de las partes estáticas de la escena, usando la mínima memoria posible. Cuando se crea una copia de un objeto se genera un enlace a un objeto ya existente. De esta forma se disminuyen los requerimientos de memoria para el almacenamiento de la base de datos. La escala de los objetos podrá ser modificada según uno o varios ejes, para ello se deberá especificar un factor factor_escala junto con el tipo de escalado, controlado por la variable Tipo, cuyos valores pueden ser X,Y,Z,XY,XZ,YZ,XYZ, según los ejes que se ven afectados por dicho factor (al resto de ejes se aplicará factor = 1).

En la cláusula parent se especificará el nodo ascendiente del objeto creado, siendo su formato el siguiente:

padre = <nombre válido objeto> (x, y, z, rotx, roty, rotz)
<nombre válido objeto> (**euler**, x, y, z, rotx, roty, rotz)
<nombre válido objeto> (**quaternio**, x, y, z, rotx, roty, rotz, rotw)

Por defecto el motor gráfico genera un único nodo **raíz** (root) del cual deben colgar el resto de nodos. Para añadir elementos a el nodo raíz se empleará como nombre del padre root. El nombre root es una palabra clave por lo cual no puede ser usada para la definición de elementos de la escena.

Los parámetros (x, y, z, rotx, roty, rotz) definen la posición y orientación del objeto respecto al nodo padre. Se definen como números reales de simple precisión (float). Con la terna (x, y, z) se define la posición del objeto hijo respecto al nodo padre. La terna (...,rotx, roty, rotz) o (euler,...,rotx, roty, rotz) asociada a los ángulos de euler determinan la orientación del sistema de referencia local del objeto respecto a la referencia del padre. En el caso de (quaternio, x, y, z, rotx, roty, rotz, rotw) los valores de (rotx,roty,rotz,rotw) determinarán las componentes de un quaternio. Es por ello que si los vértices de las geometrías están definidas con respecto a un sistema de referencia global, los valores de los parámetros anteriores deberán ser (x,y,z,rotx,roty,rotz)=(0,0,0,0,0,0).

8. macro_objetoIluminado

La macro objetoIluminado es la base del modelo de iluminación y de gestión de tramos.
Un tramo esta dividido en Interior y Exterior. Para que el motor gráfico tenga constancia de las divisiones en tramos se ha de proceder de la siguiente manera:

Se genera un grupo *'A'* que contiene otros dos grupos *'I'* y *'E'* en los cuales su nombre obligatoriamente ha de empezar con las palabras 'Interiores_' y 'Exteriores_'

```
ObjetoIluminado { _nodo_'A' }
```

nodo'A' , es el grupo del cual cuelgan los dos nodos: 'Interiores_' y 'Exteriores_'
La macro objeto iluminado analizara el grupo 'A' del que cuelgan los grupos de Interiores y Exteriores y asignara las propiedades adecuadas a cada elemento dependiendo de si es interior o exterior.

9. macro_paneltexto

La Macro paneltexto permite la inserción de textos en geometrías. La especificación de la macro de Panel de Texto es la siguiente:

```
paneltexto{nombre_panel,grupo_que_cuelga,"texto", posx,posy, tamaño, r,g,b, tipo_de_letra}
```

grupo_que_cuelga ,nombre del grupo del cual panel de texto es hijo.
Nombre_de_Panel , nombre definido en modelado que incluye la palabra **'PanelLuminoso'**
r,g,b , color del texto.

tipo_de_letra, especifica el tipo de letra de la fuente. (ej: 'arial.ttf').Este archivo se ha de encontrar en los directorios a los que tiene acceso el motor_gráfico.

10. macro_rgb

Especifica el color del fondo cuando el cielo no esta activado.

```
Rgb { r, g, b }
```

r,g,b , color. De 0 a 1.

11. macro_shader

Permite la asignación de shaders declarados externamente a grupos y/o objetos.

```
Shader  -nombre_shader- { -nodo_afectado- , -lista_de_variables- }
```

12. macro_switch

Un objeto switch permite controlar el estado en la escena de los objetos que lo componen. El nodo en cuestión es un **grupo** del que cuelgan todos sus estados posibles. La lógica asociada al nodo permite controlar interactivamente el nodo hijo que está visible en cada momento. Su sintaxis es la siguiente:

```
switch <name> {parent,n,o1,o2,...,on}
```

Cuyos parámetros coinciden con los de los objeto de tipo LOD.

13. macro_vpoint

Define la posición del punto de vista al inicio de la simulación.

```
Vpoint {num_vpoint,x,y,z,dx,dy,dz,_movable}
```

ANEXO 6. GENENT: GENERADOR DE ENTORNOS VIRTUALES FERROVIARIOS.

1.GENENT.

GENENT es la herramienta desarrollada por la Tecnología Modular para facilitar la entrada de datos en la generación de entornos virtuales ferroviarios. Esta aplicación llama a la librería GeneradorDat.dll, que será la encarga de procesar dichos archivos de configuración generando la información que requieren todos los subsistemas involucrados en la simulación virtual. En un principio la utilidad de dicha librería se limitó a la generación de entornos ferroviarios. Esto es debido a dos rasgos que caracterizan a estos entornos:

- su unidimesionalidad: posible definición del entorno a partir de una única variable, el pk (y su relación con el espacio recorrido).
- su repetibilidad: que hace factible el empleo de una Tecnología Modular.

Sin embargo, la versatilidad de la definición estructural base así como la automatización en la generación de mallas han permitido ampliar su utilización a la generación de ciudades.

La definición estructural base consiste en una red de conexiones de diferentes tipos de nodos. Cada tipo de nodo implica un enlace geométrica y funcionalmente distinto, lo que ha permitido el pasar del entorno ferroviario (PointsNode, DiamondNode…), al entorno de ciudad (RoundNode,CutNode,CrossNode…).

2.GENERDORDAT.DLL.

Para gestionar ambos tipos de entornos, ferroviario y urbano, la librería Generadorat consta de una clase Proyecto que gestiona por un lado la información relacionada con el entorno ferroviario (Super_Red_Ferroviaria) y por otro lado la información relacionada con el entorno urbano (Super_Red_Urbana).

Cada *super red* consta de una serie de clases que gestionan la topología. La red ferroviaria estructura toda su información en LDLs (Line Destokp Layout). Cada LDL estará constituido por una lista de vías y agujas (derivadas de la clase Linea). La red urbana se encuentra estructurada en Calles. Cada Calle estará formada por un conjunto de Ramales e Intersecciones. Cada Ramal estará formado por una lista de Carriles (derivados de la clase Linea). Cada Intersección constará de una lista de Rutas vehiculares (derivadas de la clase Linea).

Cada *super red* genera toda su información geométrica a través del Módulo de Ajuste Geométrico. La clase Curva Constructiva almacena la información geométrica de todas las líneas. Por un lado gestiona la información de planta y por otro lado la información de perfil, que vienen definidas por intervalos de pks. Esta información geométrica puede venir definida en diferentes formatos: biarco, NURBS o nube de puntos.

Una vez consolidada la información geométrica se generan las poligonales de todas las líneas intervinientes en el entorno (vías, agujas, carriles y rutas) y del contorno de todas las intersecciones.

A través de la clase Entorno se gestiona toda la información relacionada con el mismo. Se revisan las familias seleccionadas, se combinan los módulos de acuerdo a las opciones de generación de entorno seleccionadas y se crea la nomenclatura.

A través de la clase Señalización se gestiona toda la información relacionada con la misma. Se configuran los semáforos de acuerdo a los estándares fijados por la Tecnología Modular, se posicionan y se genera la nomenclatura. A continuación con la ayuda de la algorítmica de posicionamiento modular se posicionan todas las geometrías. Finalmente el Módulo de exportación, genera como salida todos los archivos necesarios para los subsistemas motor gráfico y simulación.

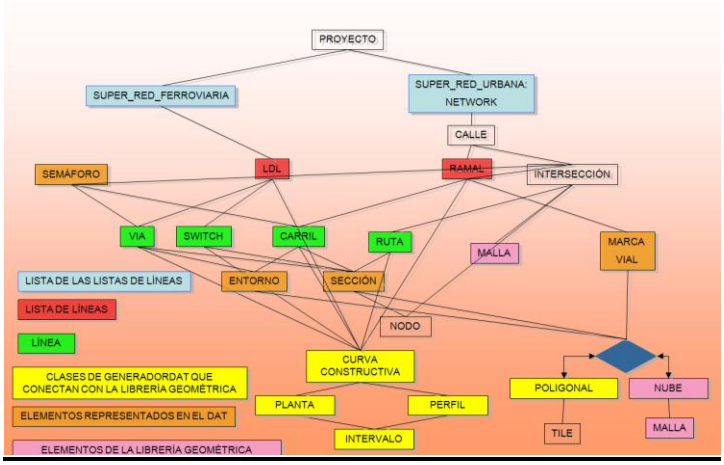


Figura A6.1.Esquema de clases topológicas empleado en la librería GeneradorDat.

A continuación se detalla paso a paso el proceso de generación de un entorno ferroviario con GENENT, desde su inicio (generación de los archivos de configuración necesarios) hasta su fin (generación de los archivos requeridos por todos los subsistemas involucrados en la simulación).

3.MENÚ CONFIGURACIÓN.

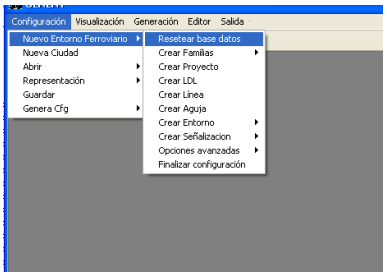


Figura A6.2.Menú Configuración de la aplicación GENENT.

3.1.GENERACIÓN DE LOS ARCHIVOS DE CONFIGURACIÓN.

La generación de un entorno virtual comienza con la definición de sus archivos de configuración. Toda la información del entorno va a ser almacenada en una base de datos: GENENT.mdb.

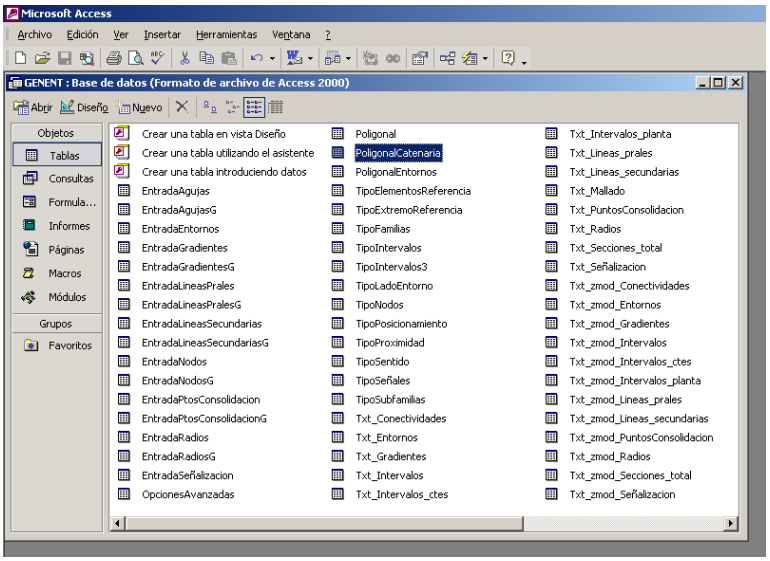


Figura A6.3.Base de datos GENENT.mdb.

Todas las tablas contenidas en la base de datos se encuentran vacías inicialmente salvo las nombradas como Tipo*.Tras la introducción de la información de partida todas las tablas Entrada* serán rellenadas y tras presionar el submenú *Finalizar configuración* se generarán las tablas Txt* y los archivos de configuración con la información de dichas tablas. Los pasos a seguir para dejar la base de datos con la información de partida son los siguientes:

1. Se comienza reseteando la base de datos GENENT.mdb con el fin de generar un entorno nuevo. La base de datos GENENT.mdb se encuentra en el mismo directorio que la aplicación.
    a. Menú Configuración: *Resetear base de datos*
2. Se continúa introduciendo todas las líneas a representar del entorno (Menú Configuración: *Crear Línea*).

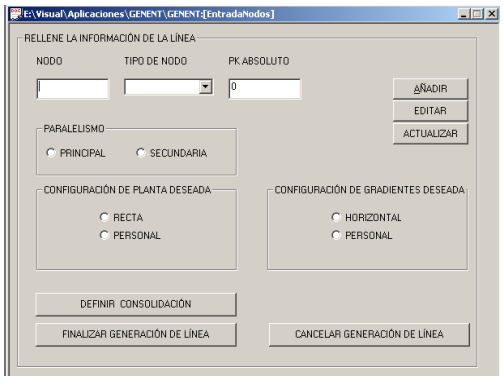


Figura A6.4.Formulario para la entrada de información de Línea en GENENT.

Para ello se introducen en orden de pks crecientes todos los nodos de la línea. En el caso de un entorno ferroviario, el nodo inicial de una línea "normal" será un TrackBoundaryNode y el de fin un TrackEndNode. Los nodos donde confluyen 3 secciones serán PointsNode (agujas y túneles de enlace). Nodos impuestos por la señalización o cualquier otro motivo ajeno al visual serán SimpleNode.

A partir del nombre y tipo de nodos se generan automáticamente las conectividades.

3. Se define si la línea es principal o secundaria y se introducen sus coordenadas de inicio:

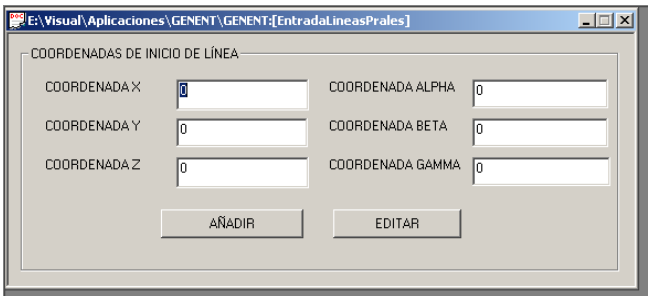


Figura A6.5.Formulario para la entrada de punto inicial de Línea en GENENT.

Si es secundaria:

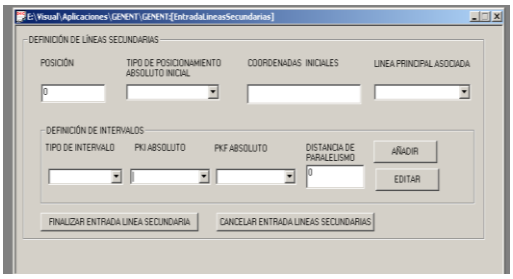


Figura A6.6.Formulario para la entrada de información de Líneas secundarias en GENENT.

Se introduce su posición respecto de la línea principal (pares a la derecha, según pks crecientes, e impares a la izquierda).Se escoge el tipo de posicionamiento inicial: pk absoluto en caso de que la línea comience con un intervalo de paralelismo, o las coordenadas cartesianas en caso de que la línea comience con un intervalo de independencia. Se elige la línea principal asociada y a continuación se definen todos los intervalos de la línea. Existen los siguientes tipos de intervalos:

a. De paralelismo, que podrán ser a la distancia estándar (3.4 metros para los proyectos de Metro de Madrid) o a otra distancia cualquiera que se introduzca.
b. De no paralelismo o independencia.
c. De conexión, se llaman así a los tramos que conectan dos zonas de paralelismo a distinta distancia a la vía ficticia.

4. Se prosigue con la elección de la planta, se deja la opción de:
   a. Planta recta
   b. Definición de radios o mezcla de radios y splines dada por el usuario. En la generación mediante spline se leerán los puntos de un topográfico y se generará la spline asociada a este conjunto de puntos (en este caso la prioridad constructiva será la topográfica). Posteriormente sobre esta spline se definirán los puntos de inserción de módulo (poligonal conjugada). En la configuración personal se podrá introducir una definición o bien toda de radios o de mezcla de splines y radios. Para cada intervalo de radio o spline se darán sus pks absolutos de inicio y fin así como su longitud. El programa posteriormente repartirá dichos radios entre las distintas secciones de la línea en función de los pks absolutos.

Figura A6.7.Formulario para la definición de la planta de una línea en GENENT.

5. A continuación se introducen los gradientes. Se tienen las mismas opciones que para la planta:
    a. Horizontal
    b. Listado de intervalos con su pk absoluto de inicio y fin.

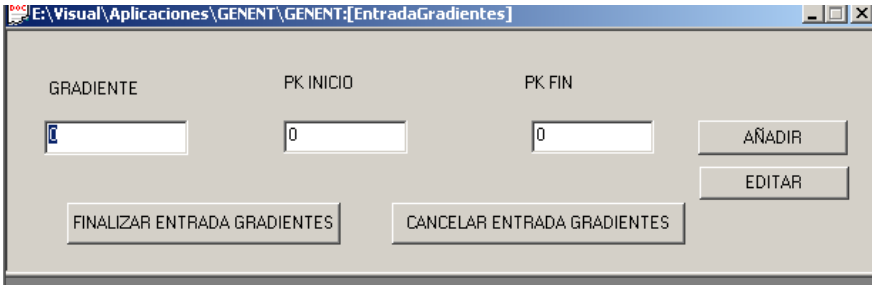


Figura A6.8.Formulario para la definición del perfil de una línea en GENENT.

6. Se añaden los puntos de consolidación de la línea, que son puntos por los cuales tendrá que pasar obligatoriamente la línea. En el caso de que no se introduzca información de puntos de consolidación podrá ocurrir:
    a. Que no exista información topográfica: no se consolida
    b. Que exista información topográfica: se consolida toda la línea.

   Se introducirán los tipos de definición geométrica deseada. Podrá ser:
    - Nube de puntos.
    - Radios.
    - Splines.

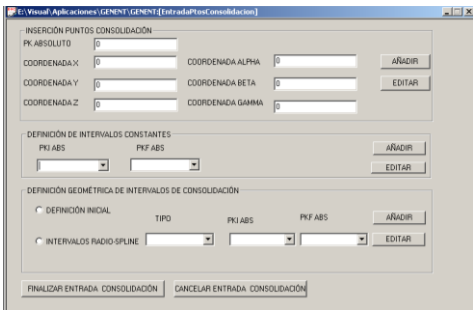


Figura A6.9.Formulario para la definición de la consolidación de una línea en GENENT.

7. El algoritmo de consolidación implica la modificación de la geometría de la línea para conseguir que ésta pase por los puntos deseados. Sin embargo pueden existir tramos donde se desee que la configuración geométrica de partida se mantenga (como es el caso

de las estaciones).Por ello se define una tabla de intervalos permanentes que nunca pueden ser tocados (las estaciones son añadidas automáticamente por el programa a esta lista de intervalos no modificables). El programa realizará una consolidación por intervalos, teniendo en cuenta los puntos por donde ha de pasar la línea y los tramos que no se pueden modificar.

8. Una vez introducida la información de todas las líneas se pasa a la introducción de las agujas. De cada aguja se selecciona su nodo de inicio y de fin de los ya existentes en la línea. Posteriormente el algoritmo de generación de agujas se encargará de darle una geometría.

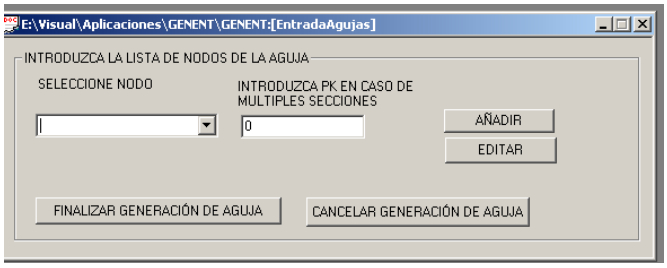


Figura A6.10.Formulario para la definición de una aguja en GENENT.

9. *Crear Entorno*: la única diferencia de posicionamiento respecto de la señalización es que existen elementos con longitud, de manera que habrá que posicionar su inicio y fin .En el caso particular de las estaciones lo habitual será situar:
    a. Su extremo inicial según pk abs y el fin respecto del inicio de estación según longitud.
    b. Ambos extremos por su pk abs.
    c. Su extremo inicial según pk abs y el fin respecto del inicio de estación según incremento de pks.

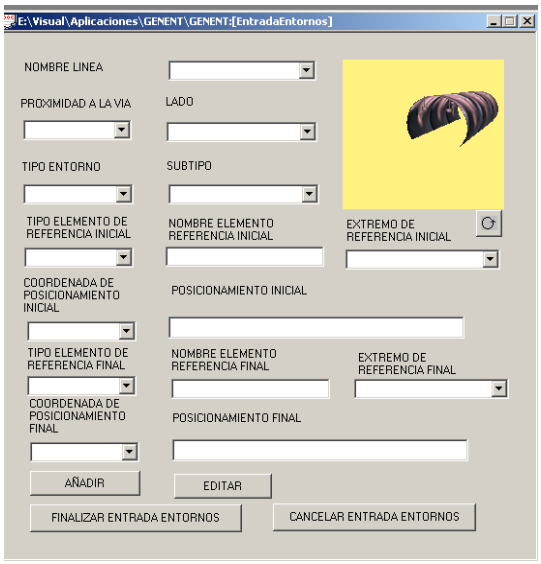


Figura A6.11.Formulario para la definición del entorno en GENENT.

10. *Crear Señalización*: Para su posicionamiento se permitirán las mismas opciones que para el entorno: un posicionamiento según pk, espacio o cartesiano. A su vez este posicionamiento podrá ser absoluto o relativo respecto a alguno de los elementos del entorno, que se suministran en el combo.

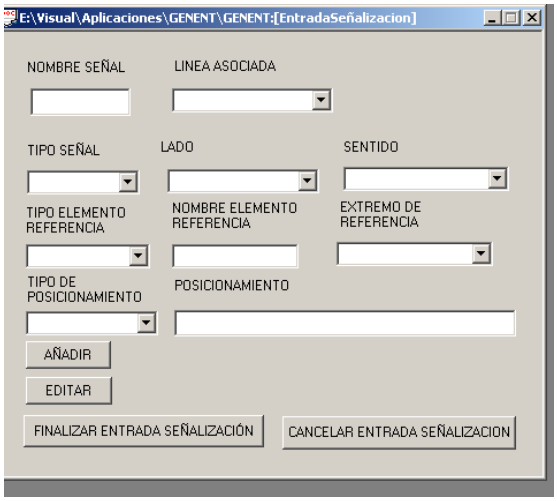


Figura A6.12.Formulario para la definición de la señalización en GENENT.

Una vez introducida la información de partida las tablas de la base de datos se encuentran en este estado:

- Toda la información de entrada se encuentra en las tablas que comienzan con "Entrada.".Las que no llevan G al final, se emplean para rellenar la información por línea a medida que el usuario la está introduciendo. Las que llevan la G al final (G de global), van guardando la información de todas las líneas. Estas tablas no serán modificadas nunca. Cualquier modificación que realiza el programa se almacenará en nuevas tablas, que tendrán el mismo nombre que las Txt * pero añadiendo la palabra z*mod*.
- Todos los archivos de configuración generados provienen de las tablas que comienzan con "Txt…"

3.2.ABRIR CONFIGURACIÓN.

Abre una base de datos de un directorio cualquiera y lo copia como GENENT.mdb en el directorio de la aplicación. Pregunta si se quiere guardar copia de la GENENT.mdb existente en ese momento en otro lugar (y así no perder la información)

3.3.GUARDAR CONFIGURACIÓN.

Esta opción se podrá emplear una vez creado el archivo DAT requerido por el motor gráfico. Realiza un guardado permanente de toda la información generada tras la completa realización de un entorno (es decir, tras haber pulsado o bien *GenerarDat_Basico* o *GenerarDat_Modificado* o bien *ReproducirDat* del menú *Generación*).

GenerarDat crea una serie de archivos txt que contienen toda la información necesaria para poder reproducir el entono ferroviario (LDL, Line Desktop Layout) que se está manejando. *Guardar* leerá estos txts y los almacenará en las siguientes tablas de GENENT.mdb:

- **Poligonal**: almacena todas las poligonales del visual.
- Por cada tabla txt_* que contiene GENENT.mdb (y que son necesarias para la creación en primera instancia de un entorno) se genera una nueva tabla con el mismo nombre pero

añadiendo z*mod* (independientemente de que se haya modificado o no la información inicial , se vuelve a generar este archivo)

- Txt_zmod_Conectividades
- Txt_zmod _Gradientes
- Txt_zmod _Radios
- Txt_zmod _Intervalos (paralelismo)
- Txt_zmod _Lineas_prales
- Txt_zmod _Lineas_secundarias
- Txt_zmod_PuntosConsolidacion
- Txt_zmod _Secciones_total
- Txt_zmod _Señalización
- Txt_zmod _Entornos
- Txt_zmod _Intervalos_ctes
- Txt_zmod _Intervalos_planta

A parte se crean las tablas:

- Txt_zmod_Mallado
- Txt_zmod_Splines

con la información generada tras la primera creación del archivo dat. La tabla Txt_zmod_Splines contiene la información de los puntos de control de la spline y la tabla Txt_zmod_Mallado los puntos de inserción y nombre del archivo que contiene la geometría 3d del modelo (*.osg) de los distintos tramos de mallado generados

La información contenida en estas tablas será la que empleen *GenerarDat modificado* y *ReproducirDat* en una futura reproducción de dicho entorno. La función *ReproducirDat* pertenece a la librería GenerarDat.dll. Esta función lo que hará será cargar un entorno, siendo la única diferencia respecto de *GenerarDat_Basico* , que las poligonales ya están creadas y hay que almacenarlas directamente en las listas correspondientes de cada sección.

4.VISUALIZACIÓN.

Nos permite visualizar los resultados obtenidos tras la introducción de una determinada configuración. De esta manera podemos comprobar que la información introducida es la deseada y retocar datos antes de continuar con el proceso global de generación del entorno. Para la generación de cada una de las vistas es necesaria la lectura de una serie de archivos. Será necesario llamar a distintas funciones de la librería GenerarDat para que todo lo necesario esté calculado.

Los datos que se pueden visualizar son los siguientes:

- Topológica: aparece una visión en planta de todas las líneas con sus secciones y todas las conexiones existentes entre ellas. Es una visión para comprobar que la definición topológica es correcta (no existe información de radios). Se llamará a la función de la librería GenerarDat.dll *Generar_topología* . Esta función crea el *Topología.txt*, que no es otra cosa que una visión esquemática, es decir, radio cero, de todas las líneas.
- Planta: representación en el plano xy de todas las líneas con la información geométrica (radios o splines). Se llamará a la función de la librería GenerarDat.dll *Generar_planta* . Esta función crea el *Planta.txt*, que no es otra cosa que la poligonal de todas las líneas con el radio dado.

- Perfil: representación en el plano pk_z de todas las líneas con la información de pendientes).Se llamará a la función de la librería GenerarDat.dll *Generar_pendientes* . Esta función crea el *Pendientes.txt*, que no es otra cosa que la poligonal de todas las líneas representando las variables pk-z.
- Señalización: representación en planta de todas las líneas con la información de señales y estaciones. Se llamará a la función de la librería GenerarDat.dll *Generar_sennalizacion*. Esta función crea el *Sennalizacion.txt*, que no es otra cosa que la poligonal de todas las líneas con el radio dado e información de estaciones y señales.

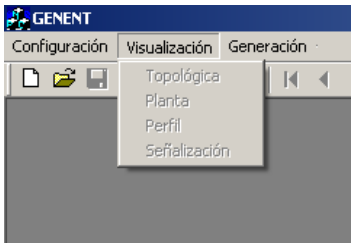


Figura A6.13.Menú de Visualización en GENENT.

5.GENERACIÓN.

El menú de Generación permite la creación de los archivos de texto plano empleados por el motor gráfico para la representación visual (el DAT). Existen diversas modalidades para la generación de estos archivos que se detallan a continuación.

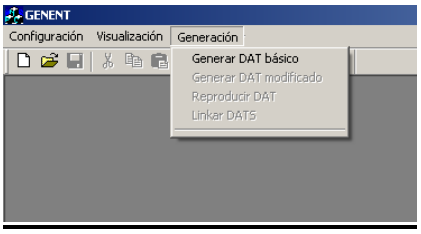


Figura A6.14.Menú de Generación en GENENT.

5.1.GENERAR DAT BÁSICO.

Llama a la función *GenerarCfgs* que crea los archivos de configuración cfg_*.txt. A continuación llama a la función *GenerarDat* de la librería GenerarDat.dll y genera por primera vez un entorno con toda la información contenida en los archivos de configuración iniciales. (tablas TXT_*).

5.2.GENERAR DAT MODIFICADO.

Llama a la función *GenerarDat* de la librería GenerarDat.dll y genera por primera vez un entorno con toda la información contenida en los archivos de configuración modificados. La

diferencia con el anterior *GenerarDat* se basa en los txts de configuración que leen de la base de datos: el *Generar Dat básico* lee las tablas TXT_* que se generan con la primera introducción de datos y el *Generar Dat modificado* genera los txts a partir de los archivos de configuración retocados en una reproducción previa del entorno (tablas TXT_zmod)*. En ambos casos se crea la poligonal de nuevo.

Para generar los archivos de configuración desde el submenú *Generar dat modificado* se llamará a la función *GenerarCfgsModificados* que leerá las tablas txt_zmod* y creará con ellas los archivos cfg*.txt

5.3.REPRODUCIR DAT.

Llama a la función *GenerarCfgsReproduccion* que crea los archivos de configuración cfg_*.txt necesarios para reproducir el dat. Estos archivos son los mismos que los generados por *GenerarCfgsModificados* a los que se añaden:

- *Cfg_poligonal*
- *Cfg_poligonal_entorno*
- *Cfg_poligonal_catenaria*
- *Cfg_mallado*
- *Cfg_splines*

Una vez disponibles los archivos de configuración, llama a la función ReproducirDat de la librería GenerarDat.dll y reproduce un entorno previamente generado leyendo para ello directamente la información de la poligonal.

5.4.LINKAR DATS.

Aparecerá un diálogo que permitirá elegir el dat inicial y final a linkar y el punto de linkado. Generará un nuevo dat con la información de todo. Es necesario evitar posibles repeticiones de nomenclatura (tanto de secciones, como elementos de señalización, entorno etc…).

5.5.FUSIONAR DATS.

Idéntico que linkar, con la diferencia de que en la fusión, las distintas partes a fusionar se convertirán en un único LDL. Se emplea en los casos en los que un LDL se tenga que generar por partes. Un ejemplo sería el caso de las líneas de metro, donde en muchas ocasiones por falta de información es necesario comenzar la generación de una línea por una zona intermedia para después generar los extremos independientemente y finalmente fusionar todo en una única línea de metro mediante un cambio de base.

6.EDICIÓN.

A través de sus diversos formularios GENENT no solo permite la generación en primera instancia de un entorno virtual sino también la edición de entornos ya generados. Dicha edición verifica en todo momento la coherencia de los cambios y la integridad de los escenarios

resultantes. Finalmente guarda la información editada en la base de datos para futuras reproducciones del entorno y genera la información necesaria para todos los subsistemas involucrados en la simulación siguiendo los estándares acordados en los Protocolos de comunicación según se explicó en los Anexos 2 y 3.

ANEXO 7. GENERACIÓN DE LA SEÑALIZACIÓN.

1.COMPONENTES DEL SISTEMA DE SEÑALIZACIÓN.

En este anexo se describe la metodología empleada por la Tecnología Modular para generar la señalización de un entorno destinado a una simulación de conducción terrestre.

Se llama bloque semafórico al conjunto de elementos físicos y funcionales constituido por un soporte y una o varias carcasas montadas sobre él. Cada tipo de soporte tiene definidas una serie de posiciones fijas en las que se pueden colocar carcasas. Cada carcasa tiene un número determinado de focos y de estados posibles de esos focos. Así pues, los bloques semafóricos se forman mediante la adecuada combinación de os siguientes componentes:

- Soporte.
- Carcasas.
- Focos.

2.SOPORTES.

Cada soporte tiene asignado un conjunto de posiciones fijas en las que pueden colocarse carcasas. A cada una de estas posiciones le corresponde un código de dos caracteres (un número y una letra). En el archivo "cfg_TipoSennalesPos_Carcasas.txt" se pueden consultar estas posiciones mediante el código que designa a cada una y obtener las coordenadas exactas de cada posición respecto al origen de coordenadas de su soporte.

Por ejemplo, si deseamos situar una carcasa en la posición "1a" del soporte "1", consultando el archivo anterior leeremos que sus coordenadas respecto al origen de coordenadas del soporte 1 son: x = 0; y = 0; z = 3m; alfa = 0; beta = 0; gama = 0.

Así pues, cuando se incorporen nuevos soportes a la bases de datos, habrán de definirse las posiciones en que se pueden colocar carcasas sobre él y añadir una línea nueva al archivo "cfg_TipoSennalesPos_Carcasas.txt" por cada una de estas posiciones.

La siguiente tabla muestra ejemplos de tipos de soportes existentes y su descripción. Su nomenclatura se formará añadiendo el prefijo "N1_" al identificador del soporte.

Identificador del soporte	Descripción	Nombre del archivo .osg
1	Columna verde de 2m	N1_1.osg
2	Columna metalizada de 2m	N1_2.osg
3	Columna verde de 2,4m	N1_3.osg
4	Columna metalizada de 2,4m	N1_4.osg
5	Báculo verde con brazo de 3,8m	N1_5.osg
6	Báculo verde con brazo de 4,3m	N1_6.osg

3.CARCASAS.

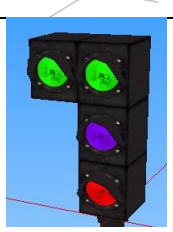
El tipo de carcasa se refiere a la forma exterior de la carcasa, pero también limita la forma y tamaño de los focos que se pueden usar en combinación con ella. En la tabla siguiente, la tercera columna indica el tipo de foco que se puede usar con cada tipo de carcasa.

El identificador de una carcasa puede tener de uno a tres caracteres [337].

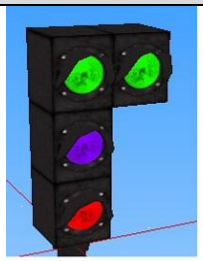
En carcasas definidas por tres letras, la primera engloba el tamaño y forma de la carcasa, la segunda letra hace referencia al color (V = verde; G = gris) y la tercera letra indica la posición del soporte según se mira a la carcasa de frente (D = derecha; I = izquierda; A = atrás; S = sin anclaje; P = apantallada).

A la hora de situar los focos será necesario usar el archivo "cfg_TipoSennalesPos_Focos_Carcasas.txt", que contiene, para cada tipo de carcasa y número de focos de la misma, las coordenadas de cada uno de los focos respecto del origen de coordenadas de la carcasa.

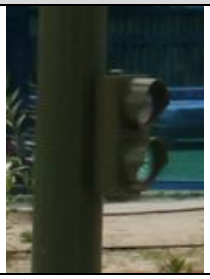
Por ejemplo, si se desea situar los focos de una carcasa tipo A de 3 focos, en este archivo veremos que las coordenadas del primer foco son: x = 0.114; y = 0; z = 0.525m; alfa = 0; beta = 0; gama = 0. Las del segundo son: x = 0.114; y = 0; z = 0.335m; alfa = 0; beta = 0; gama = 0. Las del tercero son: x = 0.114; y = 0; z = 0.145m; alfa = 0; beta = 0; gama = 0, siempre respecto del origen de coordenadas de la carcasa.

Identificador de la carcasa	Descripción	Forma y tamaño de focos que ha de usarse	Imagen
A	Una columna de focos	Circulares. Diámetro: 200mm	
B	Dos columnas de focos: La primera de una fila y la siguiente de varias	Circulares. Diámetro: 200mm	

[337] La identificación mediante un solo carácter corresponde a una nomenclatura de carcasas antigua. La incorporación de escenarios urbanos supuso la creación de una gran variedad de grupos semafóricos nuevos que hizo necesaria la definición de una nomenclatura más compleja.

Identificador de la carcasa	Descripción	Forma y tamaño de focos que ha de usarse	Imagen
C	Dos columnas de focos: La primera de varias filas y la siguiente de una	Circulares. Diámetro: 200mm	
D	Carcasa triangular	Circulares. Diámetro: 200mm	
E	Dos columnas de focos del mismo número de filas	Circulares. Diámetro: 200mm	
F	Carcasa viarios altos	Cuadrados. Lado: 200mm	
G	Carcasa de peatones gris sin anclaje	Cuadrados. Lado: 200mm	
H	Carcasa circular pequeña gris sin anclaje	2	

Identificador de la carcasa	Descripción	Forma y tamaño de focos que ha de usarse	Imagen
GGA	Carcasa de peatones gris con anclaje hacia atrás	Cuadrados. Lado: 200mm	
GGI	Carcasa de peatones gris con anclaje hacia la izquierda	Cuadrados. Lado: 200mm	
GGD	Carcasa de peatones gris con anclaje hacia la derecha	Cuadrados. Lado: 200mm	
GVS	Carcasa de peatones verde sin anclaje	Cuadrados. Lado: 200mm	
GVA	Carcasa de peatones verde con anclaje hacia atrás	Cuadrados. Lado: 200mm	
GVI	Carcasa de peatones verde con anclaje hacia la izquierda	Cuadrados. Lado: 200mm	
GVD	Carcasa de peatones verde con anclaje hacia la derecha	Cuadrados. Lado: 200mm	
HGS	Carcasa circular pequeña gris sin anclaje	Circulares. Diámetro: 100mm	

Identificador de la carcasa	Descripción	Forma y tamaño de focos que ha de usarse	Imagen
HVS	Carcasa circular pequeña verde sin anclaje	Circulares. Diámetro: 100mm	
IGS	Carcasa circular grande gris sin anclaje	Circulares. Diámetro: 200mm	El interior de las viseras en negro
IGA	Carcasa circular grande gris con anclaje hacia atrás	Circulares. Diámetro: 200mm	 El interior de las viseras en negro
IGI	Carcasa circular grande gris con anclaje hacia la izquierda	Circulares. Diámetro: 200mm	El interior de las viseras en negro
IGD	Carcasa circular grande gris con anclaje hacia la derecha	Circulares. Diámetro: 200mm	El interior de las viseras en negro
IVS	Carcasa circular grande verde sin anclaje	Circulares. Diámetro: 200mm	El interior y el exterior de las viseras en negro
IVA	Carcasa circular grande verde con anclaje hacia atrás	Circulares. Diámetro: 200mm	El interior y el exterior de las viseras en negro

Identificador de la carcasa	Descripción	Forma y tamaño de focos que ha de usarse	Imagen
IVI	Carcasa circular grande verde con anclaje hacia la izquierda	Circulares. Diámetro: 200mm	El interior y el exterior de las viseras en negro
IVD	Carcasa circular grande verde con anclaje hacia la derecha	Circulares. Diámetro: 200mm	El interior y el exterior de las viseras en negro
IVP	Carcasa circular grande verde, apantallada y con anclaje superior	Circulares. Diámetro: 200mm	 El interior y el exterior de las viseras en negro

4.FOCOS.

Los focos de un semáforo tienen tres características principales. Una de ellas es la forma o perímetro del foco. En la siguiente tabla se muestran las formas de foco existentes:

Identificador de la forma del foco	Forma del foco
1	Foco cuadrado
2	Foco redondo
3	Foco rectangular

Otra característica es el color de fondo, no el color del foco en sí, si no el del cerco a su alrededor, que suele ser del mismo material y color que la propia carcasa. En la siguiente tabla se incluyen los fondos de foco existentes:

Identificador del fondo del foco	Fondo del foco
V	Fondo de foco verde
G	Fondo de foco gris
N	Fondo de foco negro

Sin embargo, la característica principal (por ser funcional) de un foco es la imagen o el color que representa. En la siguiente tabla se indican los colores de foco existentes:

Identificador del color del foco	Descripción del foco	Estados posibles	Estado apagado propio
01	Rojo.	APAGADO; ENCENDIDO	No
02	Verde.	APAGADO; ENCENDIDO	No
03	Blanco	APAGADO; ENCENDIDO	No
04	Amarillo	APAGADO; ENCENDIDO	No
05	Morado	APAGADO; ENCENDIDO	No
06	M-Roja.	APAGADO; ENCENDIDO	Sí
07	Flecha hacia arriba	APAGADO; ENCENDIDO	Sí
08	Flecha diagonal hacia arriba a la derecha	APAGADO; ENCENDIDO	Sí
09	Flecha diagonal hacia arriba a la izquierda	APAGADO; ENCENDIDO	Sí
10	Contador numérico	APAGADO	Sí
11	Raya horizontal (Cerrado).	APAGADO; ENCENDIDO	Sí
12	Raya Vertical (Abierto).	APAGADO; ENCENDIDO; PARPADEA	Sí
13	Desvío a la derecha (┌).	APAGADO; ENCENDIDO	Sí
14	Desvío a la izquierda (┐).	APAGADO; ENCENDIDO	Sí
15	Selección de desvío entre Recto y Derecha (│ ┌).	APAGADO; VERTICAL; DRCHA	Sí
16	Selección de desvío entre Recto e izquierda (┐ │).	APAGADO; IZQDA; VERTICAL	Sí
17	Selección de desvío entre izquierda y derecha (┐ ┌).	APAGADO; IZQDA; DRCHA	Sí
18	Selección de desvío entre recto, derecha e izquierda (┐ │ ┌).	APAGADO; IZQDA; VERTICAL; DRCHA	Sí
19	Letra P.	APAGADO; ENCENDIDO; PARPADEA	Sí
20	Letra R.	APAGADO; ENCENDIDO	Sí

Identificador del color del foco	Descripción del foco	Estados posibles	Estado apagado propio
21	Viario_Rojo	APAGADO; ENCENDIDO	No
22	Viario_Naranja	APAGADO; ENCENDIDO; PARPADEA	No
23	Viario_Verde	APAGADO; ENCENDIDO	No
24	Peaton_Para	APAGADO; ENCENDIDO	Sí
25	Peaton_Pasa	APAGADO; ENCENDIDO	Sí
26	Triángulo	APAGADO; ENCENDIDO; PARPADEA	Sí
27	Raya vertical	APAGADO; ENCENDIDO; PARPADEA	Sí
28	Raya horizontal	APAGADO; ENCENDIDO	Sí

Cada color de foco existente tiene un identificador que es un número de dos dígitos.

Esta información sobre los distintos tipos de focos existentes y los posibles estados de cada uno está almacenada en el archivo "cfg_TipoSennalesEstadoFocos.txt". El estado "PARPADEA" se referirá siempre a un cambio alternativo de apariencia entre los estados "APAGADO" y "ENCENDIDO", por lo que no se podrá dar en focos en los que haya mas de una opción de Encendido, como por ejemplo los de color 10, 15, 16, 17 o 18."

Los focos 01, 02, 03, 04, 05, 21, 22, 23 tienen la misma apariencia en el estado "APAGADO", por lo que usan el mismo archivo para representar ese estado.

A continuación se describen las distintas nomenclaturas empleadas.

5.NOMENCLATURA.

5.1.BLOQUES SEMAFÓRICOS.

Un bloque semafórico se nombrará mediante una consecución de caracteres que se forma de la siguiente manera:

1. un carácter que es el identificador del soporte.
2. un número que es el número de carcasas que hay sobre el soporte.
3. los identificadores de las carcasas usadas, uno detrás de otro, sin espacios ni guiones. Cada uno de ellos puede tener uno o tres caracteres.

Identificador del soporte	Número de carcasas	Identificador de la carcasa 1	Identificador de la carcasa 2	…
Un carácter	Un número	Uno o tres caracteres	Uno o tres caracteres	…

Por ejemplo, el bloque semafórico 32IVSGGI consiste en un soporte de tipo 3, que es una columna verde de 2,4m sobre la que van montadas dos carcasas: una es la IVS (carcasa circular grande verde sin anclaje), y la otra es la GGI (carcasa de peatones gris con anclaje hacia la izquierda). Conviene señalar que en la nomenclatura de un bloque semafórico no está contenida toda la información sobre el mismo. Falta por definir en que posiciones están colocadas las carcasas sobre el soporte y cuales son los focos que lleva cada carcasa.

5.2.CARCASAS.

Una carcasa se nombra mediante la siguiente consecución de caracteres:

- un número que define los focos que hay en la carcasa.
- uno o tres caracteres que son el identificador de la carcasa.
- un carácter que es el identificador de la forma de los focos (todos los focos han de tener la misma forma: cuadrados, redondos o rectangulares).
- un carácter que es el identificador del fondo de los focos (todos los focos han de tener el mismo fondo).
- los identificadores del color de los focos, uno detrás de otro, sin espacios ni guiones.

Por ejemplo, la carcasa 3IGS2G212223 es una carcasa grande gris sin anclaje, con 3 focos redondos con fondo gris que son el viario rojo, naranja y verde.

Nº de focos	Id de la carcasa	Id de la forma de los focos	Id del fondo de los focos	Id del color del foco 1	Id del color del foco 2	Id del color del foco 3
Un número (3)	Uno o tres caracteres (IGS)	Un carácter (2)	Un carácter (G)	Un número de dos dígitos (21)	Un número de dos dígitos (22)	Un número de dos dígitos (23)

5.3.FOCOS.

El archivo que contiene un foco se nombra añadiendo el prefijo “N1_a_b_” al identificador del foco. Siendo a la forma del foco, b el color del fondo del foco.

El archivo que hace referencia al estado encendido para los focos que comparten el estado apagado es el archivo N1_a_b_c.osg”. Siendo a la forma del foco, b el color del fondo del foco y siendo c el identificador del foco.

El archivo que contiene la información del estado apagado común a varios focos, del mismo tipo, es:

“N1_a_b_0.osg”. Siendo a la forma del foco, b el color del fondo del foco.

Los focos que tienen un estado apagado específico, llevan asociado un sufijo al nombre, siendo “_0” para el estado apagado y “_1” para el estado encendido. Resultando los nombres de los ficheros N1_a_b_c_0.osg y N1_a_b_c_1.

6.FICHEROS DE REFERENCIA.

Existen una serie de ficheros que contienen la información sobre:

- los tipos de soportes de semáforo existentes y las posibles posiciones de las carcasas sobre ellos (“cfg_TipoSennalesPos_Carcasas.txt”);
- los tipos de carcasas existentes y las posibles posiciones de los focos sobre ellos (“cfg_TipoSennalesPos_Focos_Carcasas.txt”);
- los tipos de focos existentes y los estados posibles de cada uno (“cfg_TipoSennalesEstadoFocos.txt”).

También existen una serie de archivos de referencia que establecen los tipos de señales distintas que la librería GeneradorDat es capaz de procesar. Estos archivos son “cfg_TipoSennales.txt”, “cfg_TipoSubSennales.txt” y “cfg_TipoMarcasViales.txt”.

A continuación se describen cada uno de estos archivos en detalle.

6.1.ARCHIVO DE TIPOS DE SOPORTES.

Contiene la información sobre los tipos de soportes existentes y las posibles posiciones de las carcasas sobre ellos. Lleva el nombre “cfg_TipoSennalesPos_Carcasas.txt”. Este archivo tiene una fila por cada posición posible de una carcasa en cada tipo de soporte. Cada fila de este archivo contiene una serie de campos separados por tabuladores. Estos campos son los siguientes:

- Identificador del soporte (un carácter).
- Identificador de la posición de la carcasa sobre el soporte que se está describiendo (dos caracteres, un número y una letra).
- Coordenada x de la posición que se describe respecto al origen de coordenadas del soporte en cuestión.
- Coordenada y de la posición que se describe respecto al origen de coordenadas del soporte en cuestión.
- Coordenada z de la posición que se describe respecto al origen de coordenadas del soporte en cuestión.
- Coordenada alfa de la posición que se describe respecto al sistema de coordenadas del soporte en cuestión.
- Coordenada beta de la posición que se describe respecto al sistema de coordenadas del soporte en cuestión.
- Coordenada gama de la posición que se describe respecto al sistema de coordenadas del soporte en cuestión.

6.2.ARCHIVO DE TIPOS DE CARCASAS.

Contiene la información sobre los tipos de carcasas existentes y las posibles posiciones de los focos sobre ellas. Lleva el nombre “cfg_TipoSennalesPos_Focos_Carcasas.txt”. Este archivo tiene una fila por cada posición posible de un foco en cada tipo de carcasa. Hay tipos de carcasas que pueden funcionar con distinto número de focos; si es así, el archivo ha de incluir las posiciones de los focos en cada una de estas posibilidades. Cada fila de este archivo contiene una serie de campos separados por tabuladores. Estos campos son los siguientes:

- Identificador de la carcasa (uno o tres caracteres).

- Número de focos.
- Orden del foco cuya posición se describe (será un número que irá desde uno hasta el número de focos).
- Coordenada x de la posición que se describe respecto al origen de coordenadas de la carcasa en cuestión.
- Coordenada y de la posición que se describe respecto al origen de coordenadas de la carcasa en cuestión.
- Coordenada z de la posición que se describe respecto al origen de coordenadas de la carcasa en cuestión.
- Coordenada alfa de la posición que se describe respecto al sistema de coordenadas de la carcasa en cuestión.
- Coordenada beta de la posición que se describe respecto al sistema de coordenadas de la carcasa en cuestión.
- Coordenada gama de la posición que se describe respecto al sistema de coordenadas de la carcasa en cuestión.

6.3.ARCHIVO DE TIPOS DE FOCOS.

Contiene la información sobre los tipos de focos existentes y los posibles estados que pueden tener. Lleva el nombre "cfg_TipoSennalesEstadoFocos.txt". Este archivo tiene una fila por cada tipo de foco existente. Cada fila de este archivo contiene una serie de campos separados por tabuladores. Estos campos son los siguientes:

- Identificador del color del foco (un número de dos dígitos).
- Nombre del foco.
- Lista de estados posibles, separados por punto y coma.

6.4.ARCHIVO DE TIPOS DE SEÑALES.

Contiene la información sobre los tipos de señales existentes, tanto señales ferroviarias como para el tráfico rodado, a excepción de las marcas viales, que se incluyen en un archivo aparte. Lleva el nombre "cfg_TipoSennales.txt". Este archivo tiene una fila por cada tipo de señal existente. Cada fila de este archivo contiene una serie de campos separados por tabuladores. Estos campos son los siguientes:

- Nombre del tipo de señal.
- Identificador del tipo de señal (un número entero).

Existen, de momento, 14 tipos de señales distintos. Los tipos 1,2 y del 4 al 12 son usados exclusivamente por el grupo de Simulación Ferroviaria, pues se corresponden con señales ferroviarias. El tipo 3 es usado por el grupo de Tráfico Rodado, pues se corresponde con señales para el tráfico rodado. El tipo 13 está reservado para los bloques semafóricos, de los que hacen uso ambos grupos. El tipo 14 está reservado para las señales verticales del tráfico rodado.

6.5.ARCHIVO DE SUBTIPOS DE SEÑALES.

Cuando una señal es de tipo 3 ó 14 significa que es una señal para el tráfico rodado. En este caso ha de corresponderle un subtipo de señal para describirla correctamente. Este archivo contiene la información sobre los subtipos de señales existentes. Lleva el nombre "cfg_TipoSubSennales.txt". Este archivo tiene una fila por cada subtipo de señal existente. Cada

fila de este archivo contiene una serie de campos separados por tabuladores. Estos campos son los siguientes:

- Identificador del tipo de señal (un número).
- Identificador del subtipo de señal (un número).
- Nombre del subtipo de señal.
- Lista de estados posibles del switch asociado a la señal.

6.6.ARCHIVO DE TIPOS DE MARCAS VIALES.

Este archivo contiene la información sobre los tipos de marcas viales existentes. Lleva el nombre "cfg_TipoMarcasViales.txt". Este archivo tiene una fila por cada subtipo de señal existente. Cada fila de este archivo contiene una serie de campos separados por tabuladores. Estos campos son los siguientes:

- Puede tener uno de los tres valores siguientes: "MARCAVIAL_LONGITUDINAL", "MARCAVIAL_PUNTUALES", "MARCAVIAL_TRANSVERSAL".
- El valor de este campo depende directamente del campo anterior. Dependiendo de si la marca en longitudinal, puntual o transversal, en este campo hay que poner ML, MP o MT, respectivamente.
- Nombre del tipo de marca vial.

En este momento existen los siguientes tipos de marcas viales:

Identificador de la marca vial	Longitudinal, puntual o transversal
CONTINUA_ESTANDAR	Longitudinal
DISCONTINUA_ESTANDAR	Longitudinal
DISCONTINUA_ESPECIAL	Longitudinal
CONTINUA_DOBLE	Longitudinal
STOP	Puntual
CEDA	Puntual
DETENCION	Transversal
CEDA	Transversal
PASO_CEBRA	Transversal
APARCAMIENTO	Longitudinal

7.FICHEROS DE POSICIONAMIENTO.

La librería GeneradorDat a la hora de generar un escenario concreto hace uso de una serie de archivos de configuración de ese entorno. Entre estos archivos se encuentran los que describen el posicionamiento de la señalización. Estos archivos se describen a continuación.

7.1.ARCHIVO DE POSICIONAMIENTO DE SEÑALES DEL ENTORNO: CFG_SENNALIZACION.TXT.

Por cada señal que exista en el entorno, se ha de rellenar una fila de este archivo para que GeneradorDat lo pueda procesar, salvo en dos casos:

- cuando la señal sea un bloque semafórico, habrá una fila por cada carcasa del bloque;
- cuando la señal sea una marca vial, irá incluida en el archivo de marcas viales, no en éste.

Cada fila de este archivo constará de los siguientes campos:

- Id de la señal: es un conjunto de caracteres único para cada señal (normalmente un número, aunque cualquier serie de caracteres es válida). No puede haber dos señales en el entorno con el mismo Id.
- Nombre de la señal: Suministrado hasta ahora por el subsistema de simulación. Para la librería GeneradorDat es indiferente este nombre. Ej: M/SM1. Para el caso de bloques semafóricos lo formaremos siguiendo las reglas de su nomenclatura vistas.
- Identificador del tipo de señal: número entero que identifica el tipo de señal (segundo campo del archivo cfg_TipoSennales.txt).
- Subtipo de señal: secuencia de caracteres que indica el subtipo de señal. En caso de ser un bloque semafórico, en este campo incluiremos la nomenclatura de la carcasa que estamos describiendo en esta fila del archivo, según las reglas de su nomenclatura vistas. Cuando no sea una carcasa de un bloque semafórico lo que estamos describiendo, en este campo incluiremos el número entero que identifica el subtipo de señal (segundo campo del archivo cfg_TipoSubSennales.txt).
- Nombre de la línea en la que se encuentra la señal. Cuando es una señal referida a un carril del tráfico rodado el nombre de la línea se forma a partir del nombre del carril: "L" + nodo inicial + "F" + posición carril + "_" + nodo final + "F" + posición carril; Ejemplo: LNI1F3_NT2F3).
- Lado: Si la señal se ha de colocar al lado derecho del ramal (cuando es una señal referida a un carril del tráfico rodado hablamos del ramal al que pertenece ese carril) o línea (cuando es una señal ferroviaria), ponemos una “D”, si es al lado izquierdo una “I”.
- Sentido: Este campo indica hacia qué sentido del ramal o línea ha de estar orientada la señal: “C” indica hacia el sentido de los Pks crecientes y “D” hacia el sentido de los Pks decrecientes.
- Tipo de elemento de referencia. Es el elemento del entorno de simulación que se usa como referencia para situar la señal. Puede ser: “Linea Principal” (si se da el Pk absoluto), “Linea”, “Seccion”, “Estacion”, “Señal” o “Entorno”.Cuando estemos tratando una señal del tráfico rodado la referenciaremos siempre a un carril del entorno de circulación, y usaremos el identificador “Linea”.
- Identificador del elemento de referencia: identificador asociado al elemento de referencia. Cuando estemos tratando una señal del tráfico rodado daremos aquí el nombre del carril que usamos de referencia. Este carril ha de ser el que esté más a la derecha o más a la izquierda (dependiendo de si la señal va en la parte izquierda o derecha de la calle) de todos los de su mismo sentido del ramal al que pertenezca.
- Extremo de referencia: Indica si es el extremo inicial (“I”) o final (“F”) del elemento de referencia el que se usa para situar la señal.
- Tipo de coordenada: Las señales se sitúan respecto a un elemento de referencia de dos formas posibles: Indicando el Pk en que se encuentra (“PK”) o la coordenada geométrica curvilínea (“S”). También se puede situar una señal dando sus coordenadas absolutas, sin usar elemento de referencia (“C”).

- Coordenada X: Si tipo de coordenada es “C”, aquí se indica la coordenada x absoluta de la señal, si no, se pone “0”.
- Coordenada Y: Si tipo de coordenada es “C”, aquí se indica la coordenada y absoluta de la señal, si no, se pone “0”.
- Coordenada Z: Si tipo de coordenada es “C”, aquí se indica la coordenada z absoluta de la señal, si no, se pone “0”.
- Coordenada ALFA: Si tipo de coordenada es “C”, aquí se indica la coordenada alfa absoluta de la señal, si no, se pone “0”.
- Coordenada BETA: Si tipo de coordenada es “C”, aquí se indica la coordenada beta absoluta de la señal, si no, se pone “0”.
- Coordenada GAMMA: Si tipo de coordenada es “C”, aquí se indica la coordenada gama absoluta de la señal, si no, se pone “0”.
- PK: Si tipo de coordenada es “PK”, aquí se indica el Pk del elemento de referencia en que se encuentra la señal, si no, se pone “0”.
- S: Si tipo de coordenada es “S”, aquí se indica la coordenada geométrica curvilínea del elemento de referencia en que se encuentra la señal, si no, se pone “0”.
- Tranviario: Cuando estemos situando una carcasa de un semáforo tranviario, ponemos aquí “1”, en caso contrario ponemos “0”.
- Código de posicionamiento de la carcasa: Cuando estemos situando una carcasa de un semáforo, aquí pondremos el código de posicionamiento de la carcasa en el soporte. En el caso contrario dejaremos este campo vacío.

7.2.ARCHIVO DE POSICIONAMIENTO DE MARCAS VIALES DEL ENTORNO: CFG_CIUDAD_MARCAS_VIALES.TXT.

Por cada marca vial que exista en el entorno, tendremos que rellenar una fila de este archivo para que la librería GeneradorDat lo pueda procesar. Cada fila de este archivo constará de los siguientes campos:

- Id de la marca vial: es un conjunto de caracteres único para cada señal (normalmente un número). No puede haber dos marcas viales en el entorno con el mismo Id.
- En este campo siempre pondremos “CurvaConstructiva”.
- Nombre de la curva constructiva del ramal en que se encuentra la marca vial. El nombre de la curva constructiva es el nombre del ramal, pero sin la “R” inicial. Ejemplo: si el ramal es RNA1_NI0 la curva constructiva es NA1_NI0.
- Carril inicial. Se refiere al número que identifica al carril dentro del ramal. Ejemplo: el carril CRNA1_NI0_3 es el carril 3.
- Carril final. Cuando se trate de una marca vial longitudinal, el carril inicial y el final son el mismo. La marca vial longitudinal se situará a la izquierda o a la derecha del carril referido dependiendo del valor del último campo de esta fila del archivo. Cuando se trate de una marca vial puntual, el carril inicial y el final son el mismo, y se refiere al carril en el que se ha de situar la marca vial puntual. Cuando se trate de una marca vial transversal, esta cubrirá desde el carril inicial hasta el final transversalmente al ramal. Si la marca vial transversal ha de situarse en ambos sentidos de circulación y existe una mediana, habrá que colocar dos marcas viales, cada una cubriendo todos los carriles de cada sentido.
- Tipo de marca vial. Puede ser “MARCAVIAL_LONGITUDINAL”, “MARCAVIAL_PUNTUALES” o “MARCAVIAL_TRANSVERSAL”, tal y como figura en el archivo de referencia “cfg_TipoMarcasViales.txt”.
- Identificador de la marca vial. Marca el tipo concreto de marca que estamos usando. Los tipos concretos de marca vial que se pueden usar también vienen definidos en el archivo de referencia “cfg_TipoMarcasViales.txt”.
- En este campo siempre pondremos “TN”.

- Pk de inicio.
- Pk de fin. Cuando se trate de una marca vial puntual o transversal, el Pk de inicio y el final son el mismo. Cuando se trate de una marca vial longitudinal, esta se situará desde el Pk inicial hasta el final. Si en el campo Pk de fin escribimos "-1", esto quiere decir que la marca llegará hasta el final del carril.
- Lado del carril. Este campo sólo tiene sentido rellenarlo o leerlo cuando se trata de una marca vial longitudinal. Es un valor booleano: un 0 indica que la marca se coloca a la izquierda (en el sentido del ramal) del carril referenciado; un 1 indica que se coloca a la derecha.

7.3.SOPORTES DE SEMÁFOROS Y POSICIONAMIENTO DE CARCASAS.

A continuación se definen las posiciones en las que se pueden colocar carcasas de semáforo en cada soporte existente. Para cada soporte existirán una serie de posiciones fijas. A cada posición le irá asociada una 'etiqueta' (un nombre).

Cada posición supone unas coordenadas y una orientación respecto al origen de coordenadas del soporte en cuestión (situado en su base).

7.3.1.SOPORTES 1 Y 2: COLUMNA DE 2M DE ALTURA.

Posibles posiciones de carcasas:

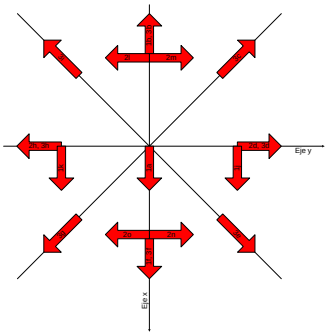


Figura A7.1Posibles posiciones de carcasas en un soporte de Tipo 1 ó 2. Croquis en planta.

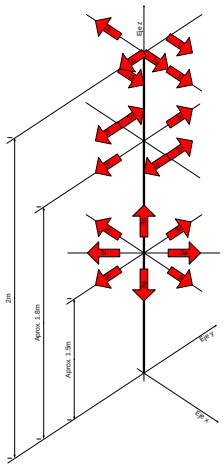


Figura A7.2Posibles posiciones de carcasas en un soporte de Tipo 1 ó 2. Croquis en vista isométrica

A continuación se muestran una serie de ejemplos:

Figura A7.3Ejemplo 1

Figura A7.4Ejemplo 2

Ejemplo 1:

A este Semáforo le corresponde:

- Una carcasa tipo A de tres focos en la posición 1a.
- Una carcasa tipo G de dos focos en la posición 2l.
- Una carcasa de pulsador de peatones en la posición 3f.

Ejemplo 2:

A este Semáforo le corresponde:

- Una carcasa tipo A de tres focos en la posición 1f.
- Una carcasa tipo A de tres focos en la posición 1b.
- Una carcasa tipo G de dos focos en la posición 2h.
- Una carcasa de pulsador de peatones en la posición 3f.

7.3.2.SOPORTES 3 Y 4: COLUMNA DE 2.4 M DE ALTURA.

Posibles posiciones de carcasas:

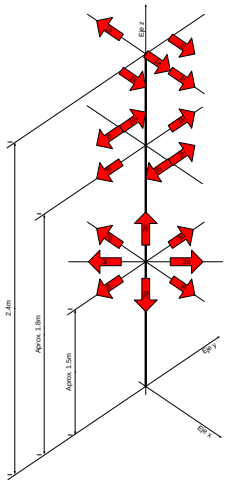


Figura A7.5Posibles posiciones de carcasas en un soporte de Tipo 3 ó 4. Croquis en vista isométrica.

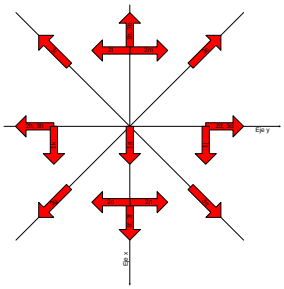


Figura A7.6Posibles posiciones de carcasas en un soporte de Tipo 3 ó 4. Croquis en plata.

7.3.3.SOPORTE 5: BÁCULO CON BRAZO DE 3.8M.

Posibles posiciones de carcasas:

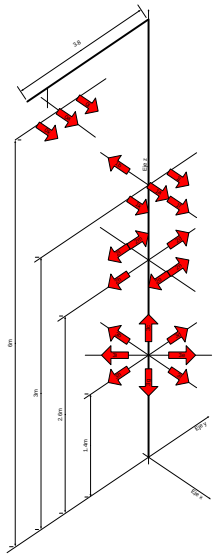


Figura A7.7Posibles posiciones de carcasas en un soporte de Tipo 5. Croquis en vista isométrica

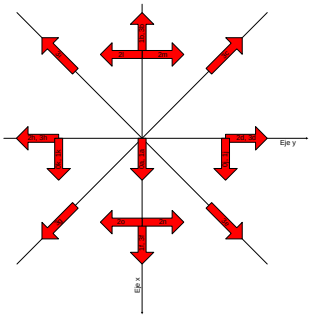


Figura A7.8Posibles posiciones de carcasas en un soporte de Tipo 5. Croquis en planta.

A continuación se muestran un ejemplo:

Figura A7.9Ejemplo.

A este Semáforo le corresponde:

- Una carcasa tipo A de tres focos en la posición 0a.
- Una carcasa tipo A de tres focos en la posición 1f.
- Una carcasa tipo G de dos focos en la posición 2l.
- Una carcasa de pulsador de peatones en la posición 3f.

7.3.4.SOPORTE 6: BÁCULO CON BRAZO DE 4.3M.

Posibles posiciones de carcasas:

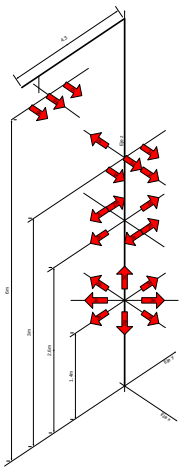


Figura A7.10Posibles posiciones de carcasas en un soporte de Tipo 6. Croquis en vista isométrica

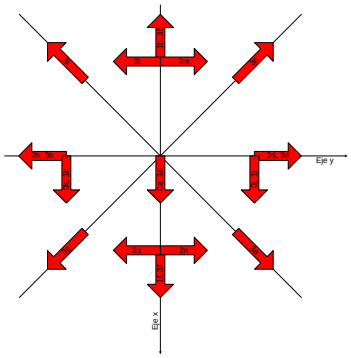


Figura A7.11Posibles posiciones de carcasas en un soporte de Tipo 6. Croquis en planta

En todos los casos vistos las coordenadas exactas de cada posición respecto al origen de coordenadas del báculo vienen especificadas en el archivo “cfg_TipoSennalesPos_Carcasas.txt. Lo más frecuente es encontrar semáforos en los que en las posiciones más altas existan carcasas tipo A (columna de focos), en las intermedias carcasas tipo G (peatones) y en las más bajas carcasas tipo H (viarios bajo) o pulsadores para los peatones.

En las posiciones 2l y 2m y 2o y 2n no puede haber carcasas simultáneamente, porque se solaparían.